Introduction to Digital Logic

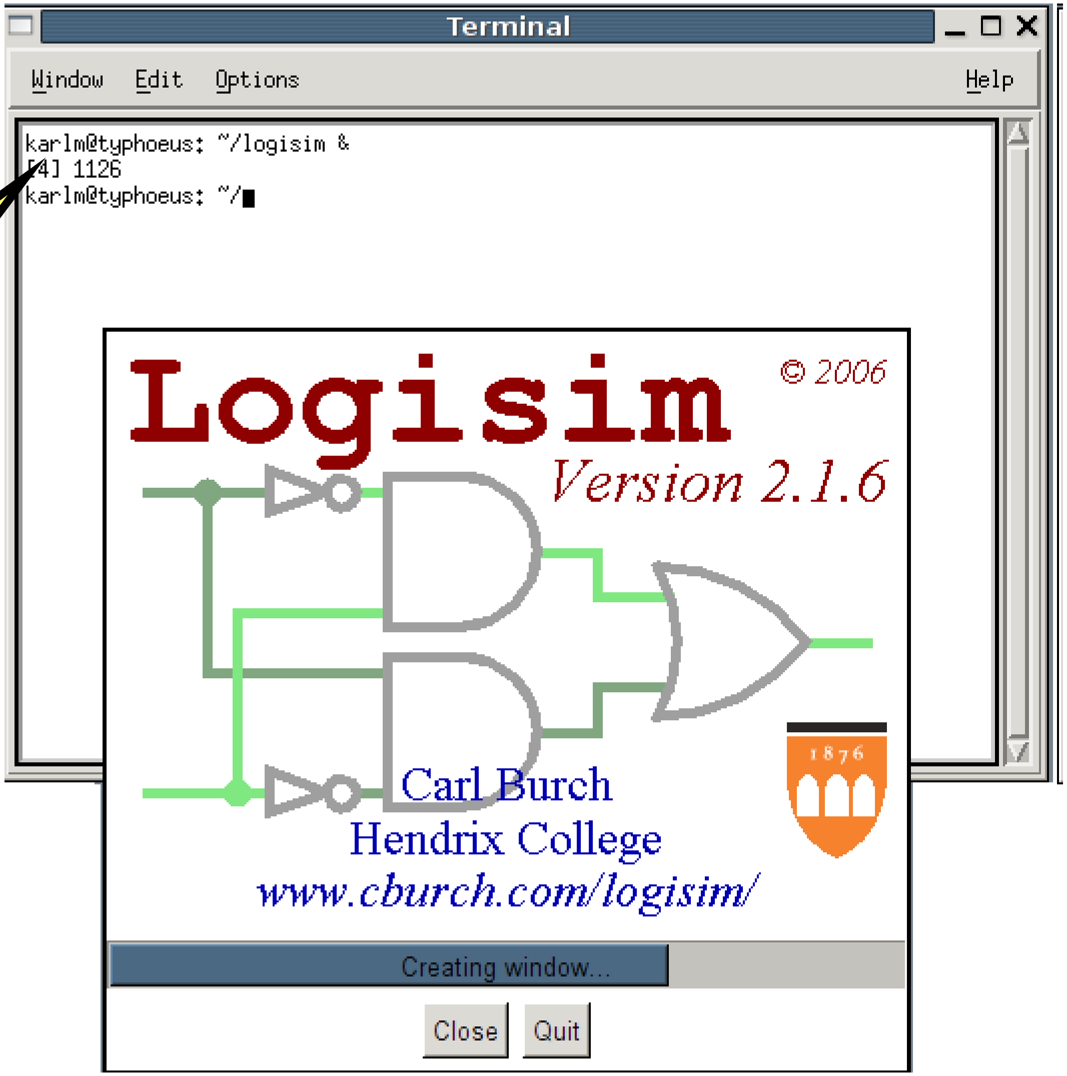
Autumn 2008

Gates, Plexers, Decoders, Registers, Addition and Comparison

karl.marklund@it.uu.se

On the IT department Unix system....

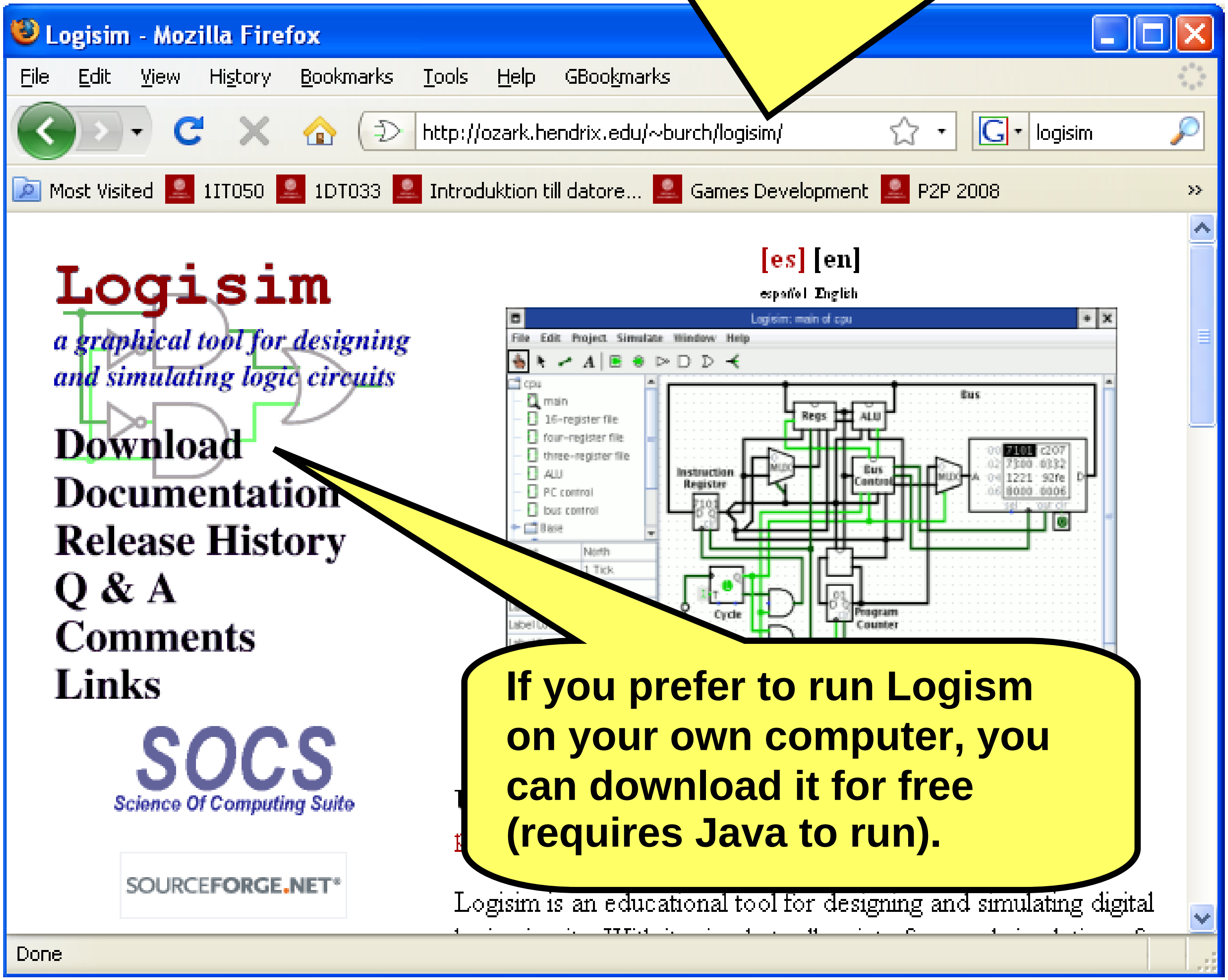
...open up a command shell and type `logisim` and press enter to start Logisim.



```
karlm@typhoeus: ~/logisim &
[4] 1126
karlm@typhoeus: ~/
```

Creating window...

http://ozark.hendrix.edu/~burch/logisim



Logisim

a graphical tool for designing and simulating logic circuits

Download
Documentation
Release History
Q & A
Comments
Links

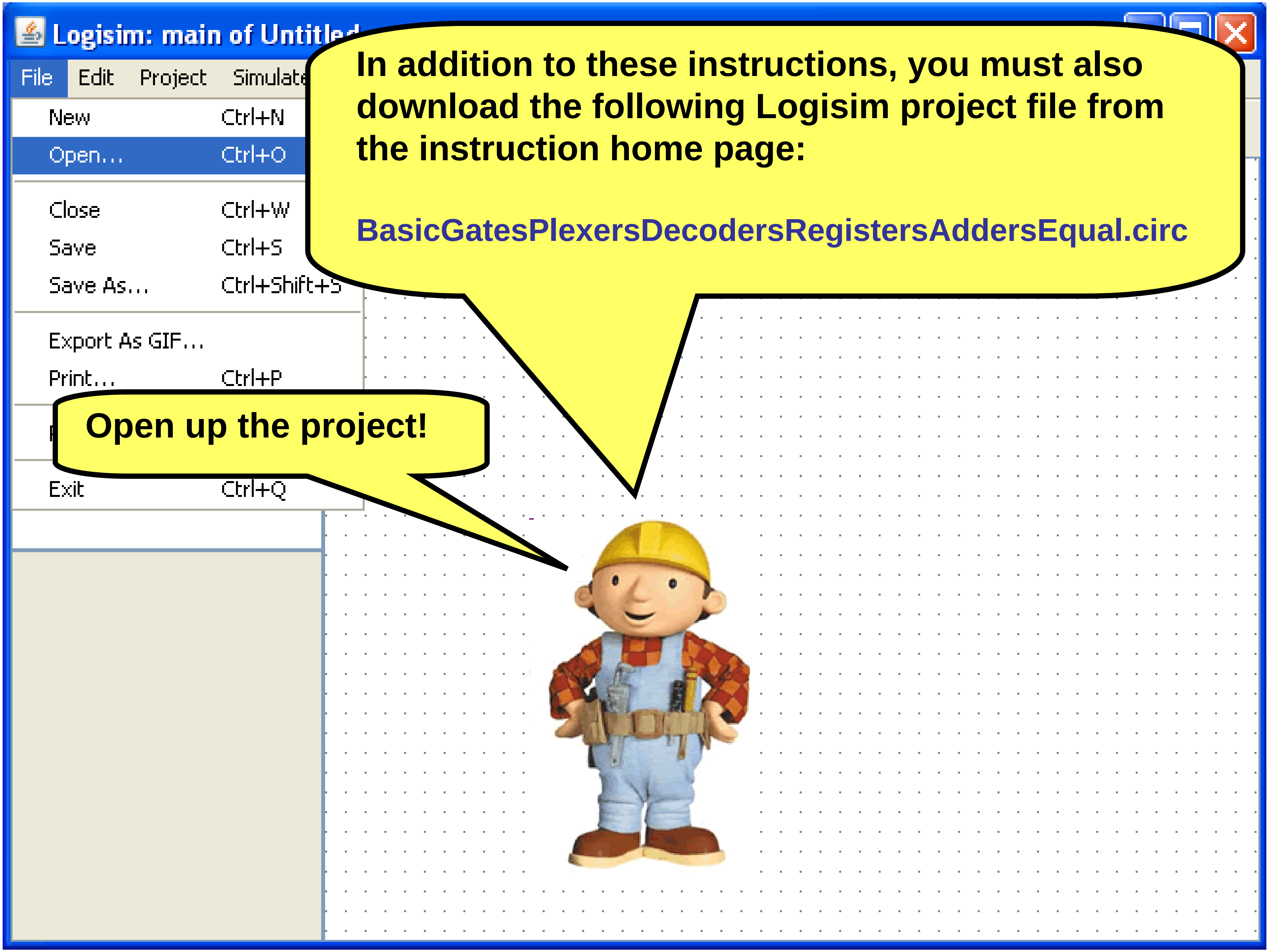
SOCS
Science Of Computing Suite

SOURCEFORGE.NET

[es] [en]
español English

Logisim is an educational tool for designing and simulating digital

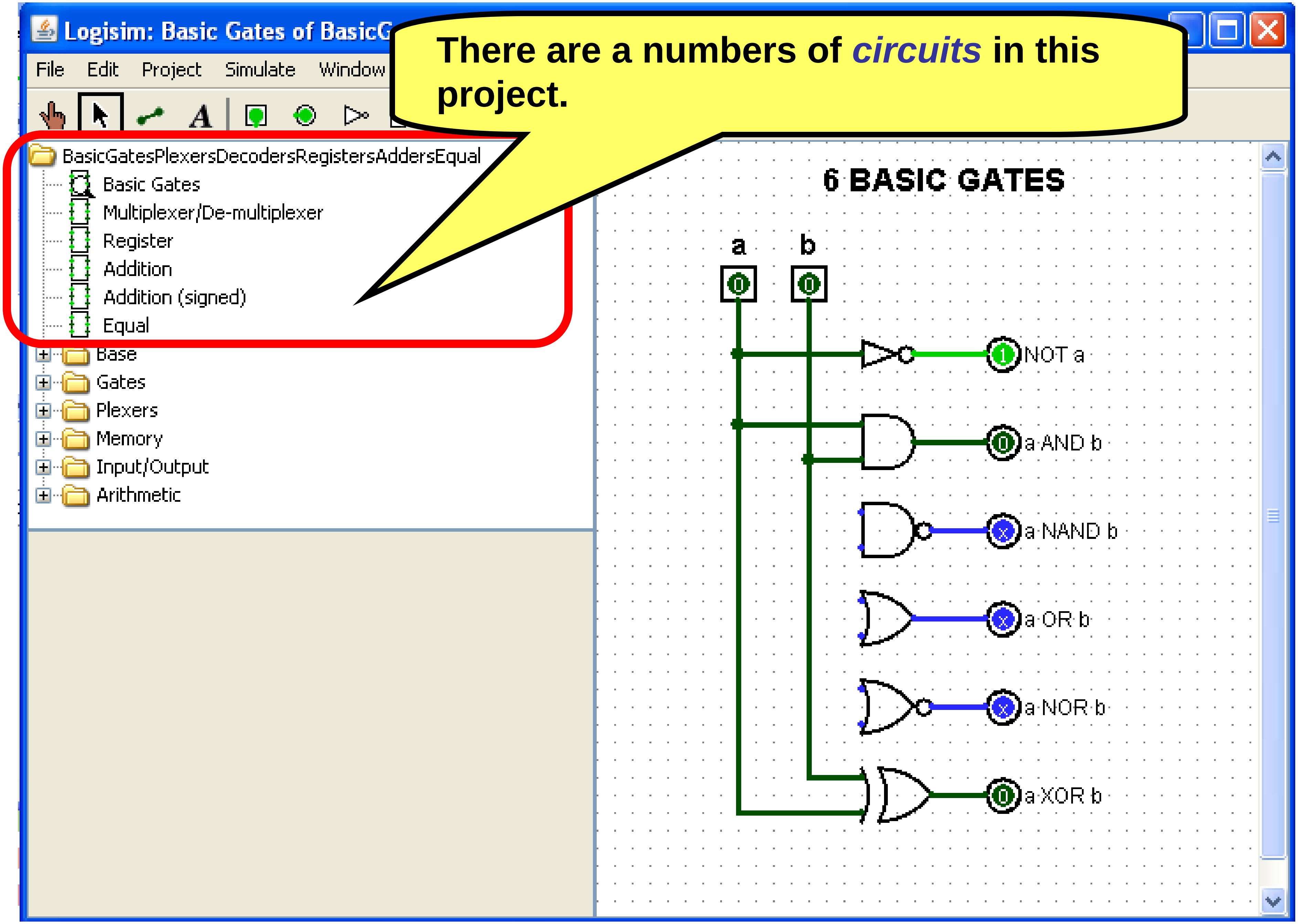
If you prefer to run Logism on your own computer, you can download it for free (requires Java to run).



In addition to these instructions, you must also download the following Logisim project file from the instruction home page:

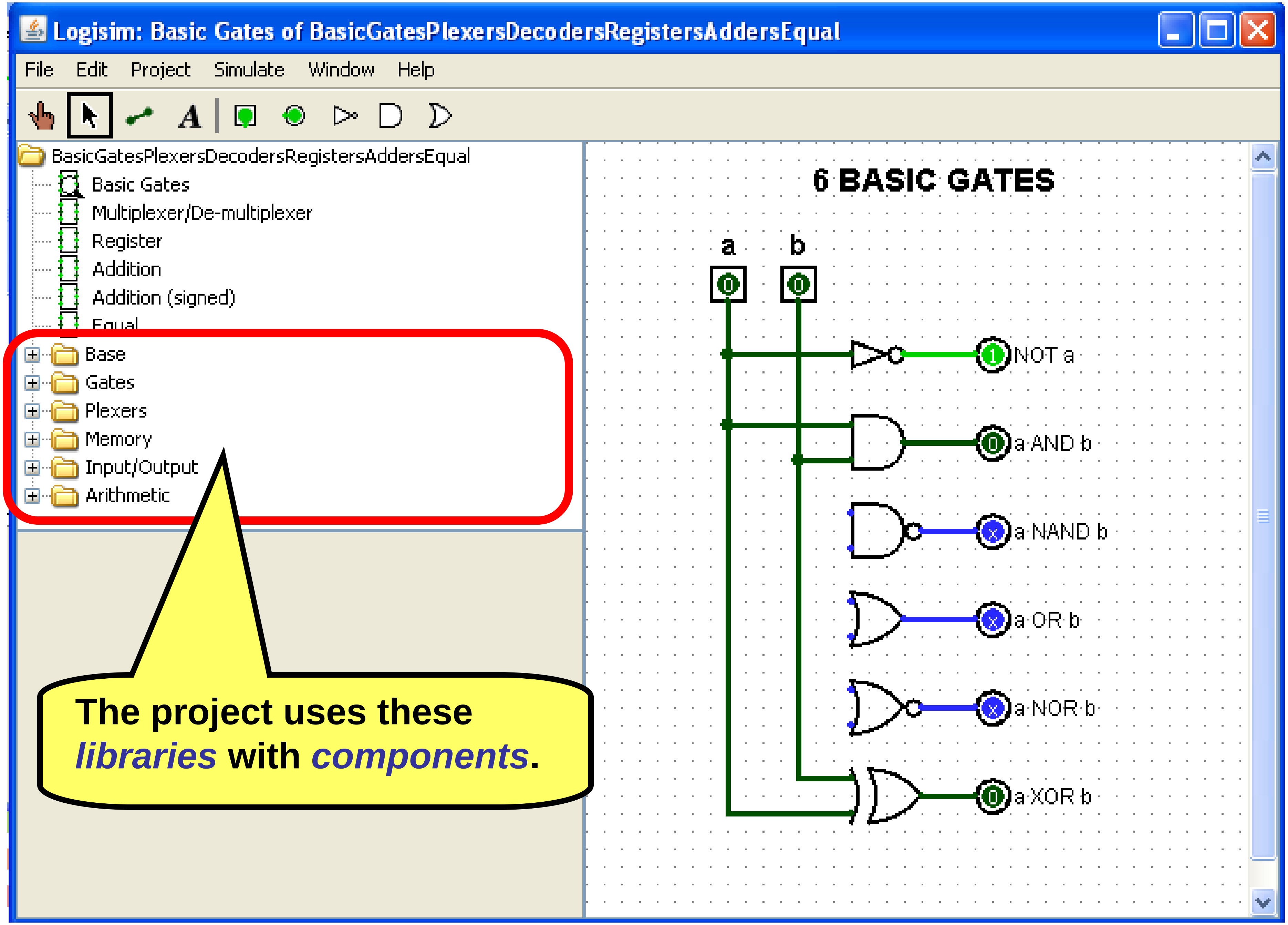
BasicGatesPlexersDecodersRegistersAddersEqual.circ

Open up the project!

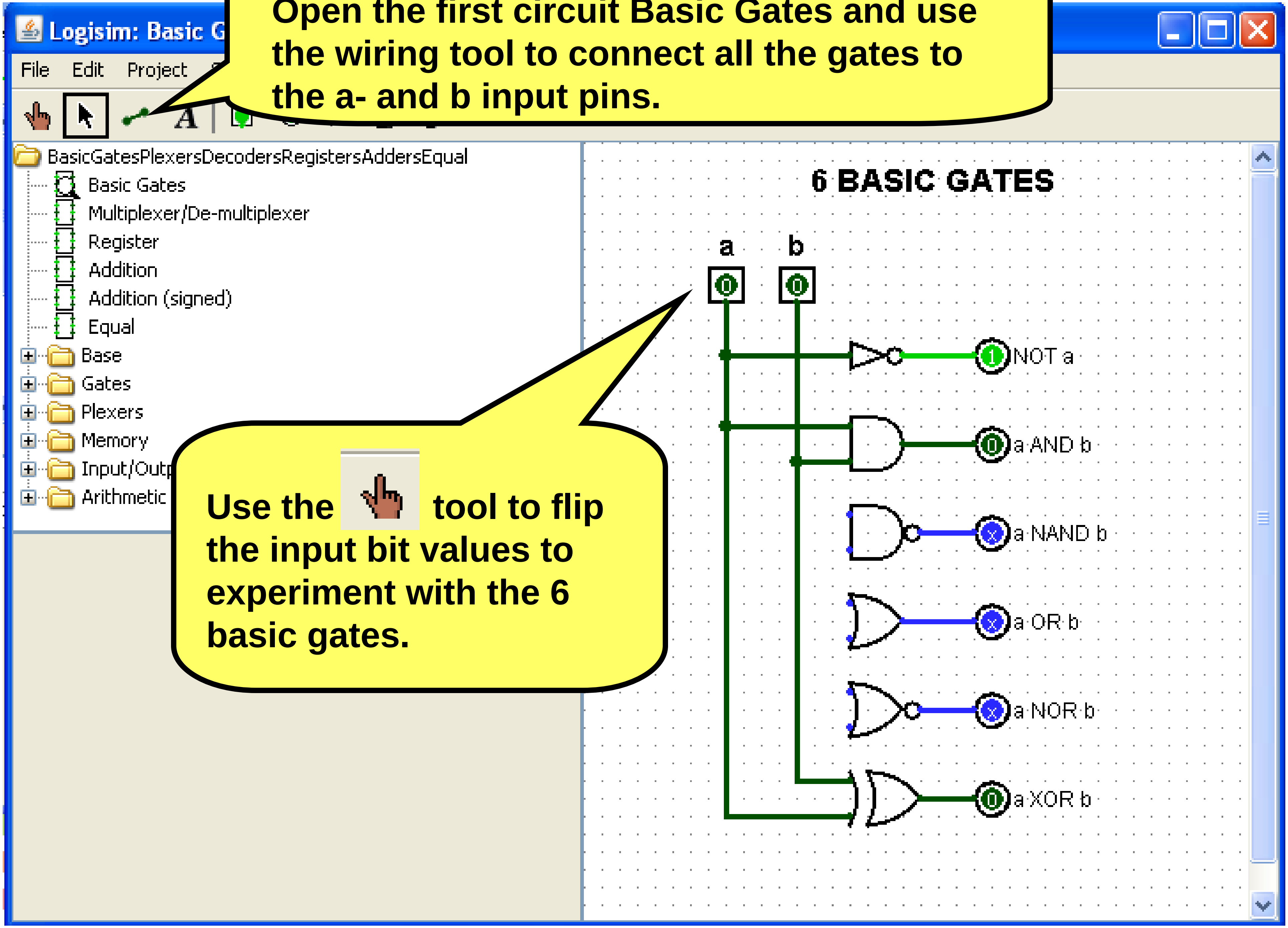


There are a numbers of *circuits* in this project.

Logisim: Basic Gates of BasicG

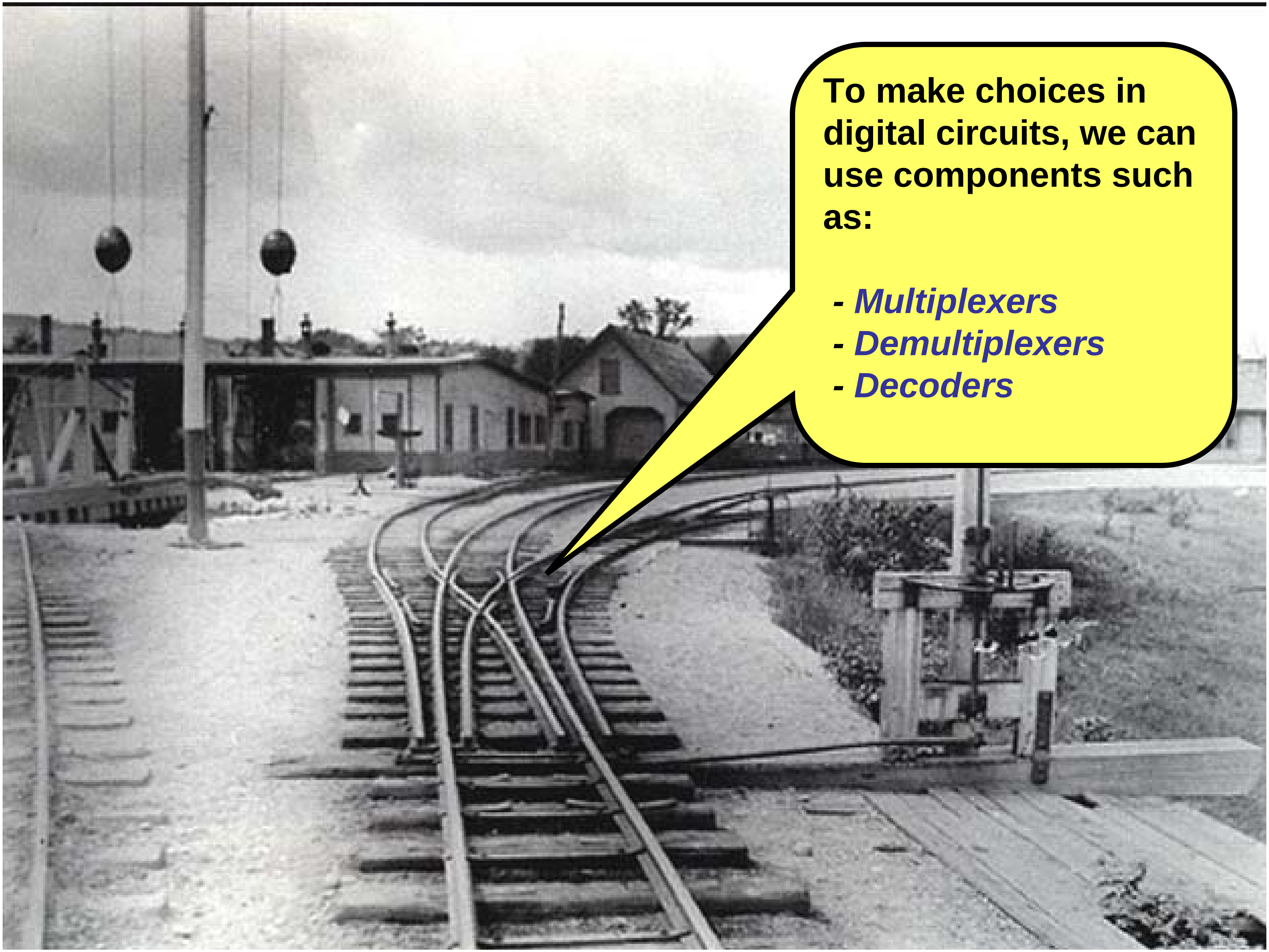

Logisim: Basic Gates of BasicGatesPlexersDecodersRegistersAddersEqual
File Edit Project Simulate Window Help
BasicGatesPlexersDecodersRegistersAddersEqual
Basic Gates
Multiplexer/De-multiplexer
Register
Addition
Addition (signed)
Equal
Base
Gates
Plexers
Memory
Input/Output
Arithmetic
6 BASIC GATES
a
b
0
0
1
NOT a
0
a AND b
x
a NAND b
x
a OR b
x
a NOR b
0
a XOR b

The project uses these *libraries* with *components*.



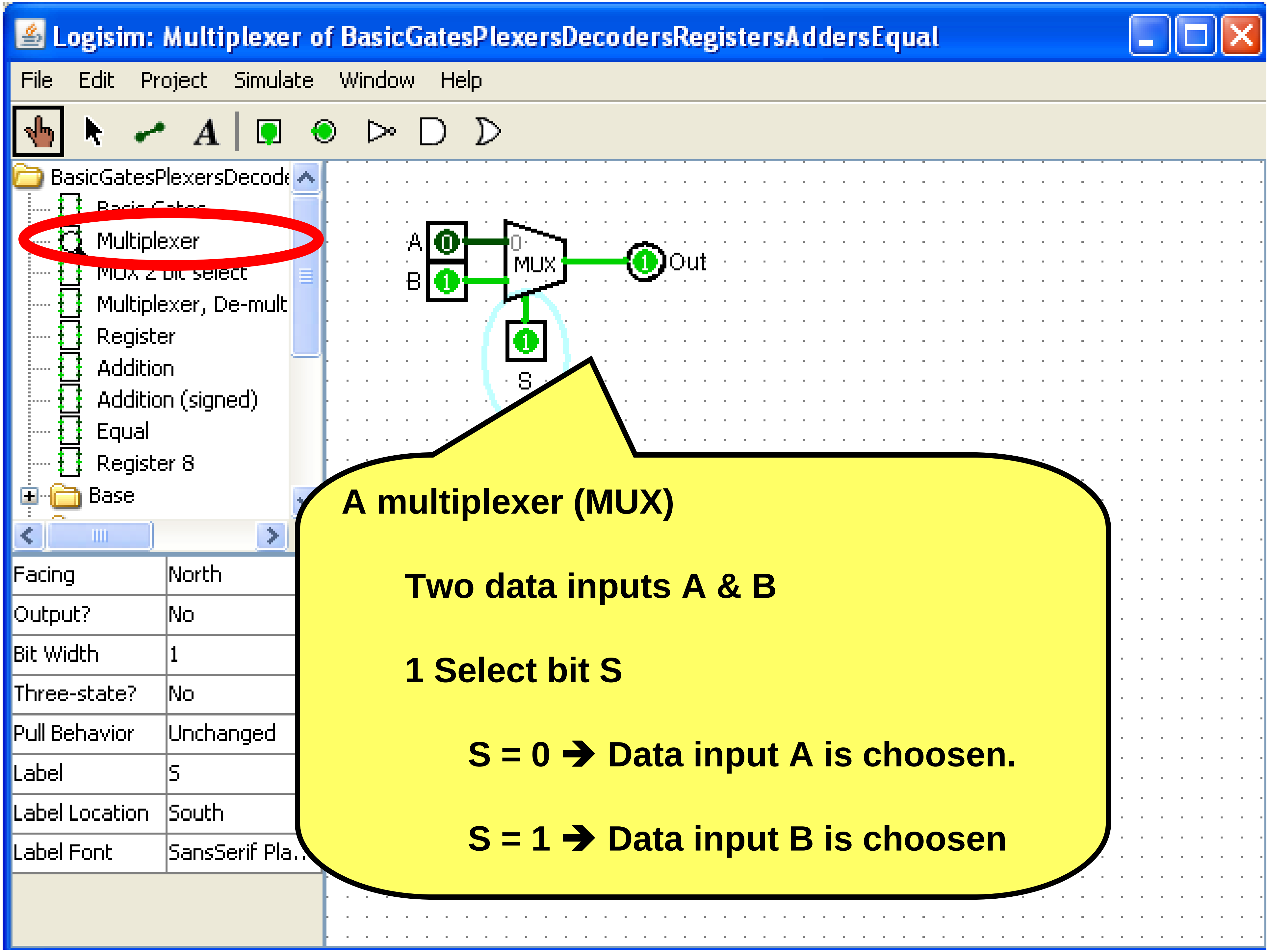
Open the first circuit Basic Gates and use the wiring tool to connect all the gates to the a- and b input pins.

Use the 👆 tool to flip the input bit values to experiment with the 6 basic gates.



To make choices in digital circuits, we can use components such as:

- ***Multiplexers***
- ***Demultiplexers***
- ***Decoders***



Facing	North
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	S
Label Location	South
Label Font	SansSerif Pla…

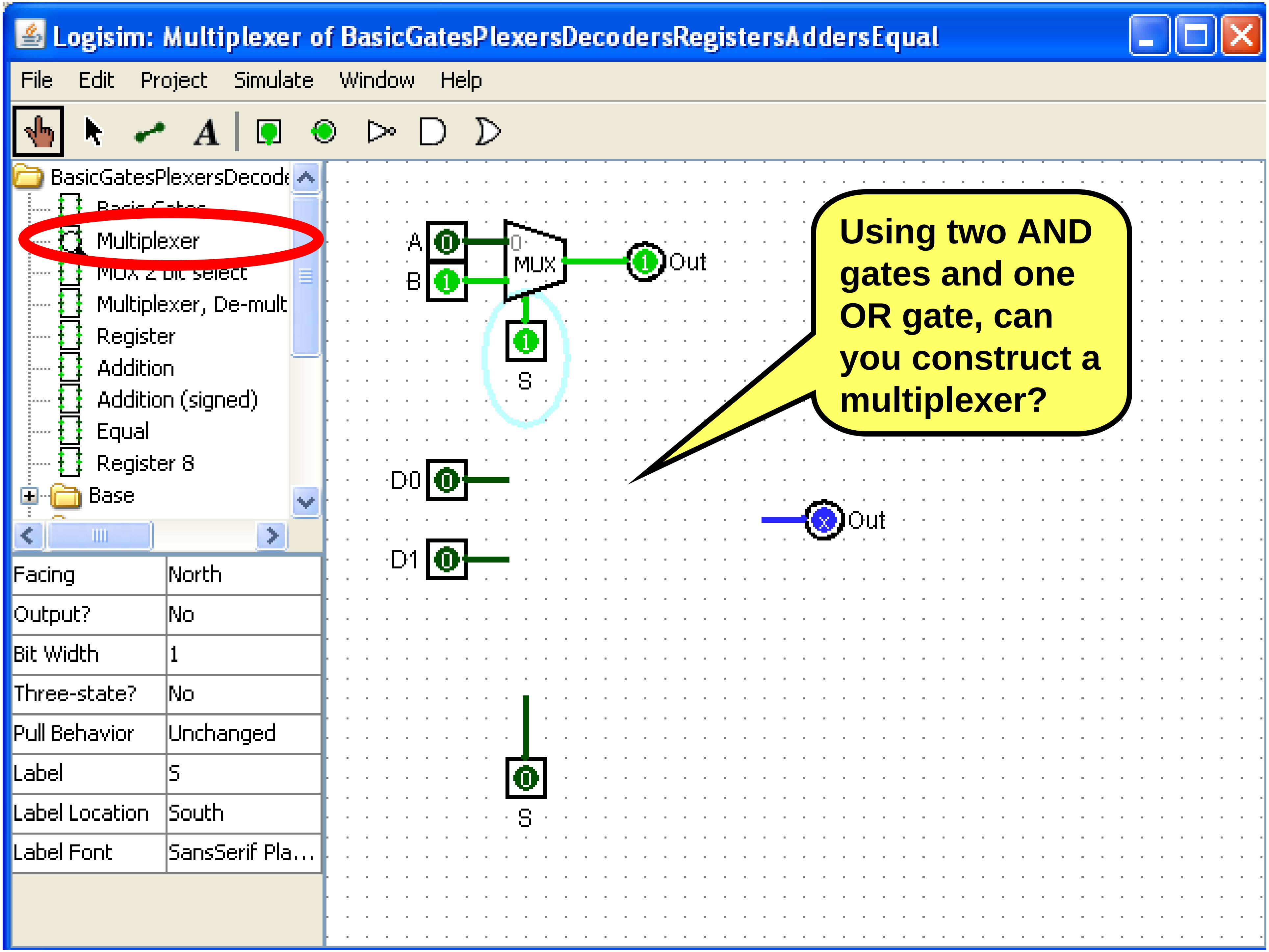
A multiplexer (MUX)

Two data inputs A & B

1 Select bit S

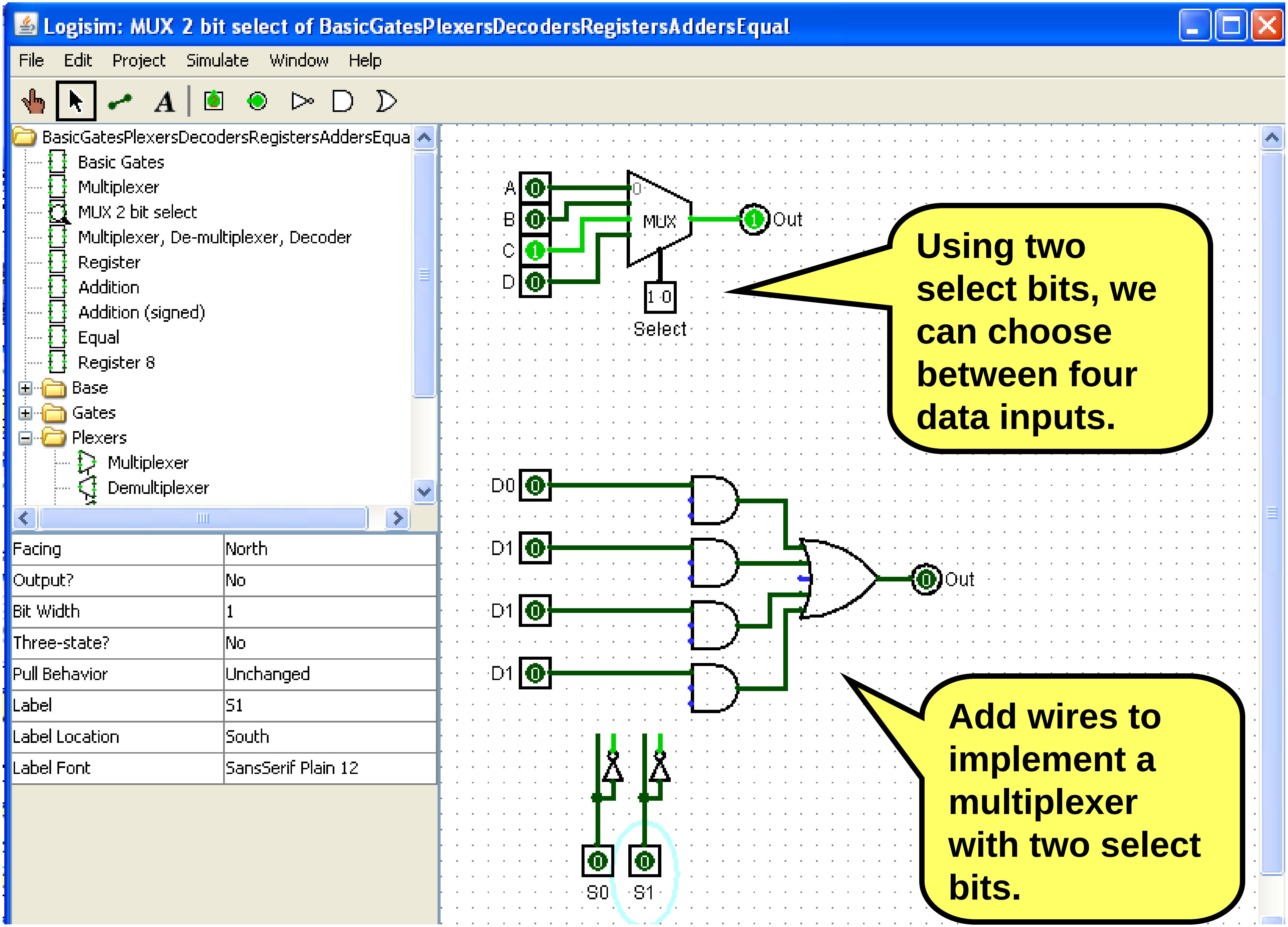
S = 0 ➔ Data input A is choosen.

S = 1 ➔ Data input B is choosen



Facing	North
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	S
Label Location	South
Label Font	SansSerif Pla…

Using two AND gates and one OR gate, can you construct a multiplexer?

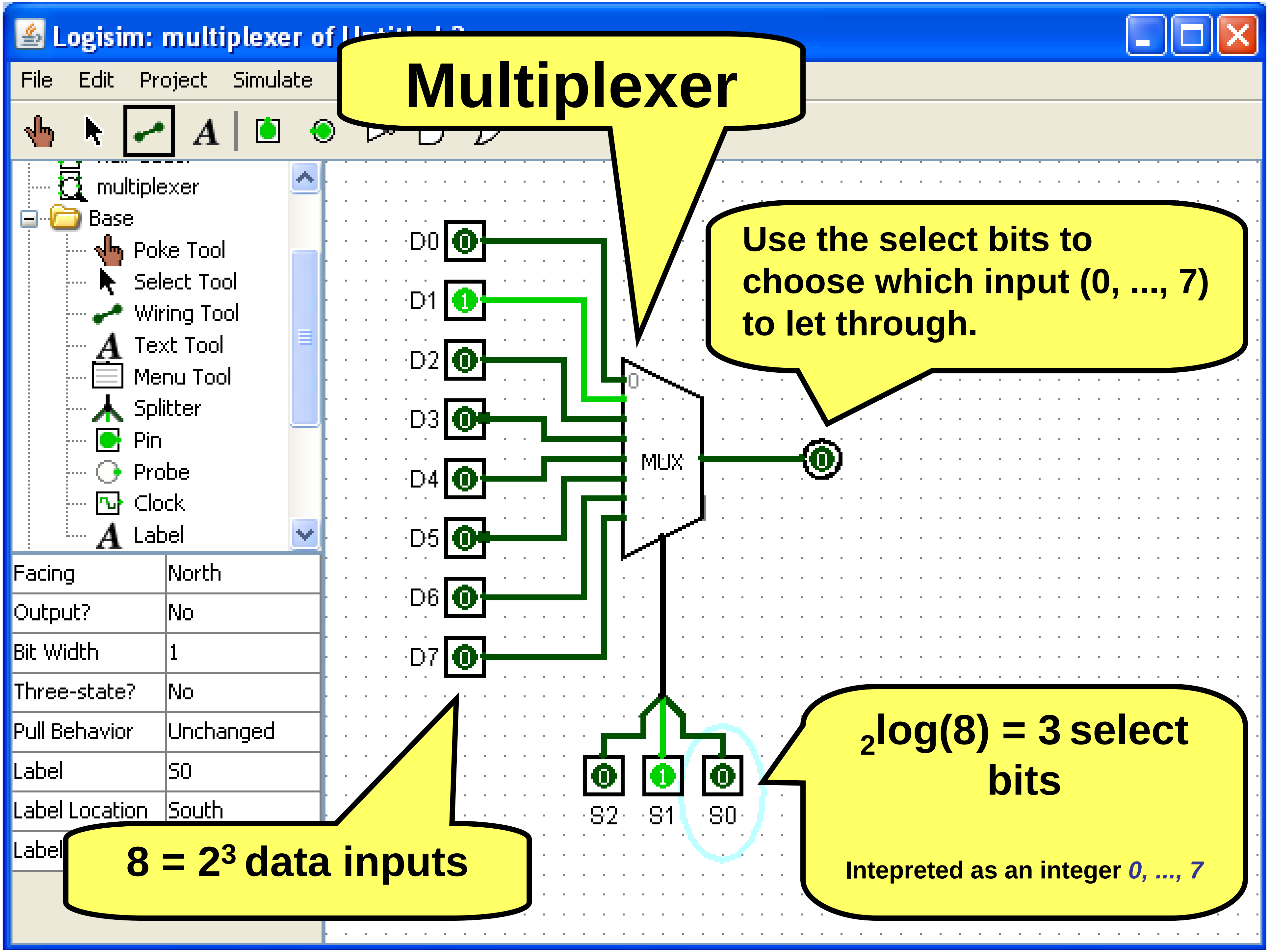


Facing	North
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	S1
Label Location	South
Label Font	SansSerif Plain 12

Using two select bits, we can choose between four data inputs.

Add wires to implement a multiplexer with two select bits.

Multiplexer



Facing	North
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	S0
Label Location	South
Label	

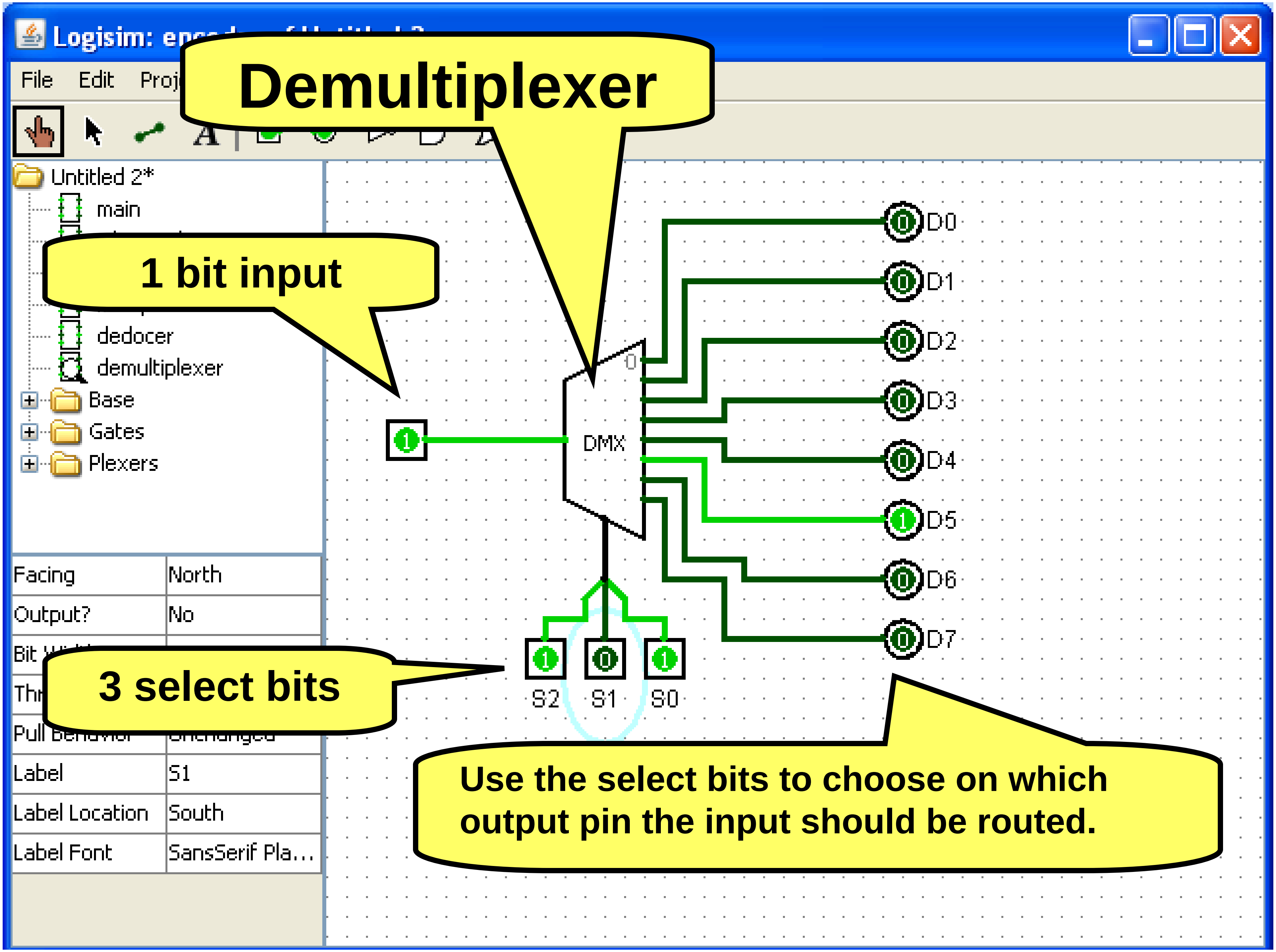
Use the select bits to choose which input (0, ..., 7) to let through.

8 = 2^3 data inputs

$_2\log(8) = 3$ select bits

Intepreted as an integer *0, ..., 7*

Demultiplexer

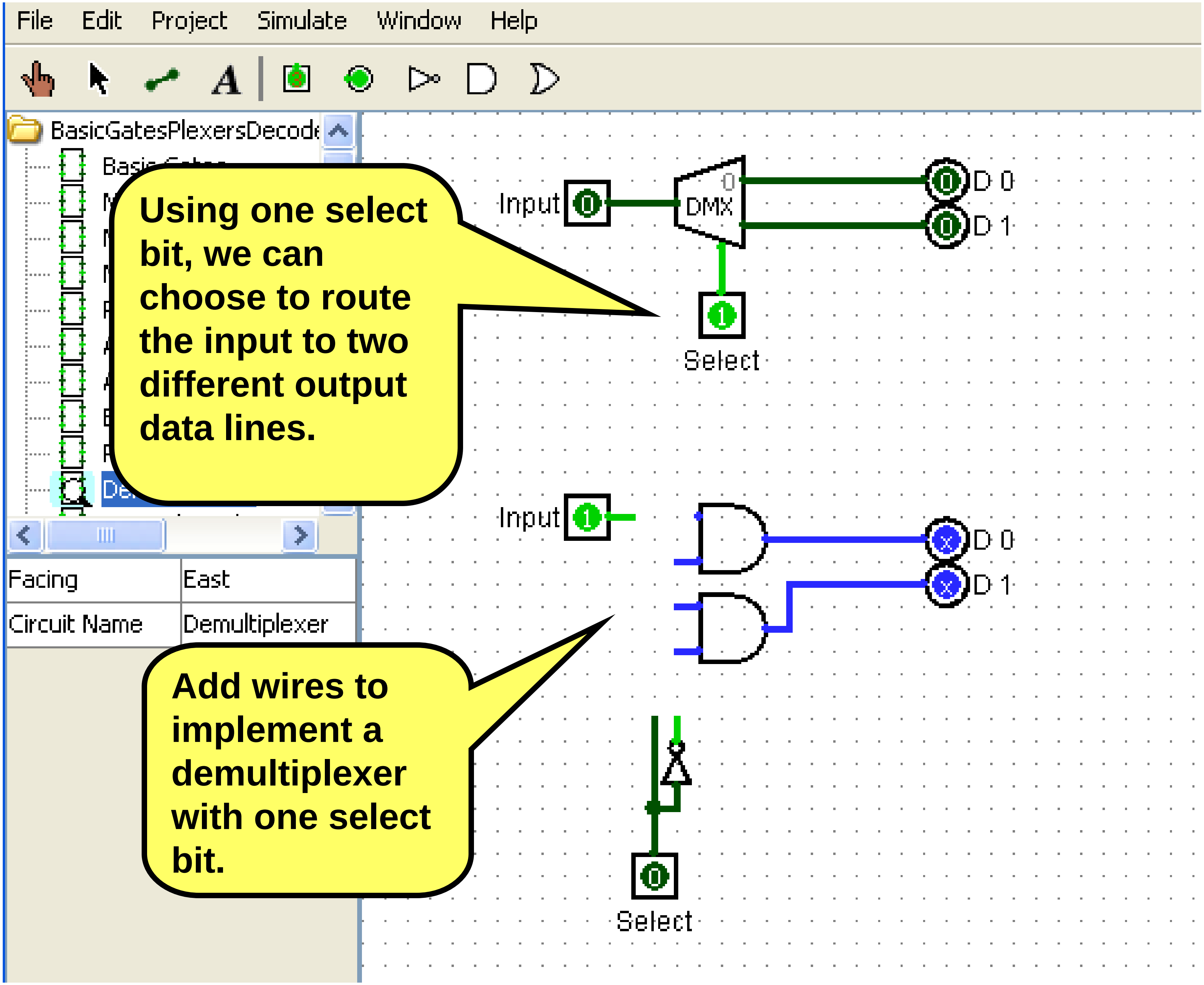


Facing	North
Output?	No
Bit W	
Thr	
Pull Behavior	Unchanged
Label	S1
Label Location	South
Label Font	SansSerif Pla...

1 bit input

3 select bits

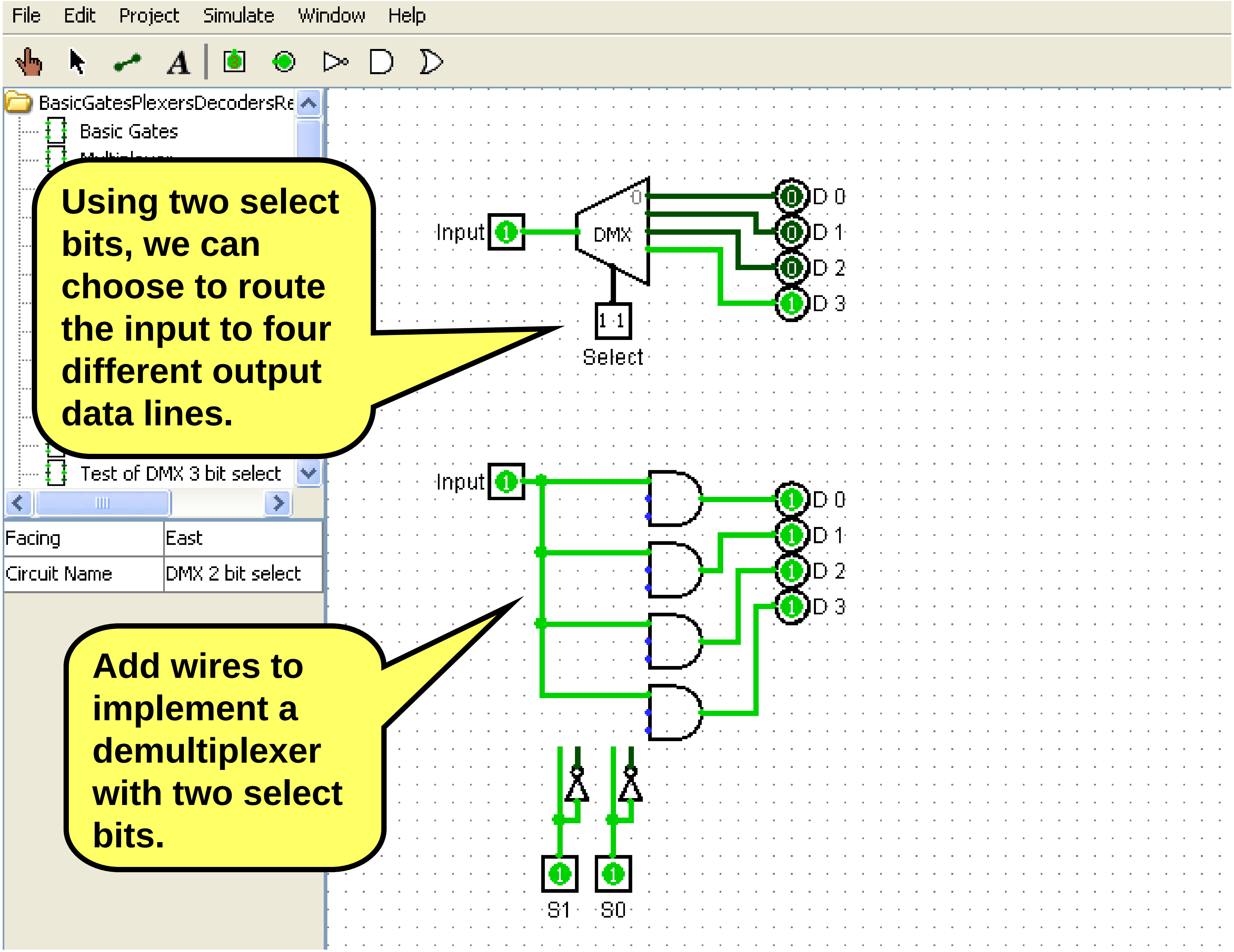
Use the select bits to choose on which output pin the input should be routed.



Facing	East
Circuit Name	Demultiplexer

Using one select bit, we can choose to route the input to two different output data lines.

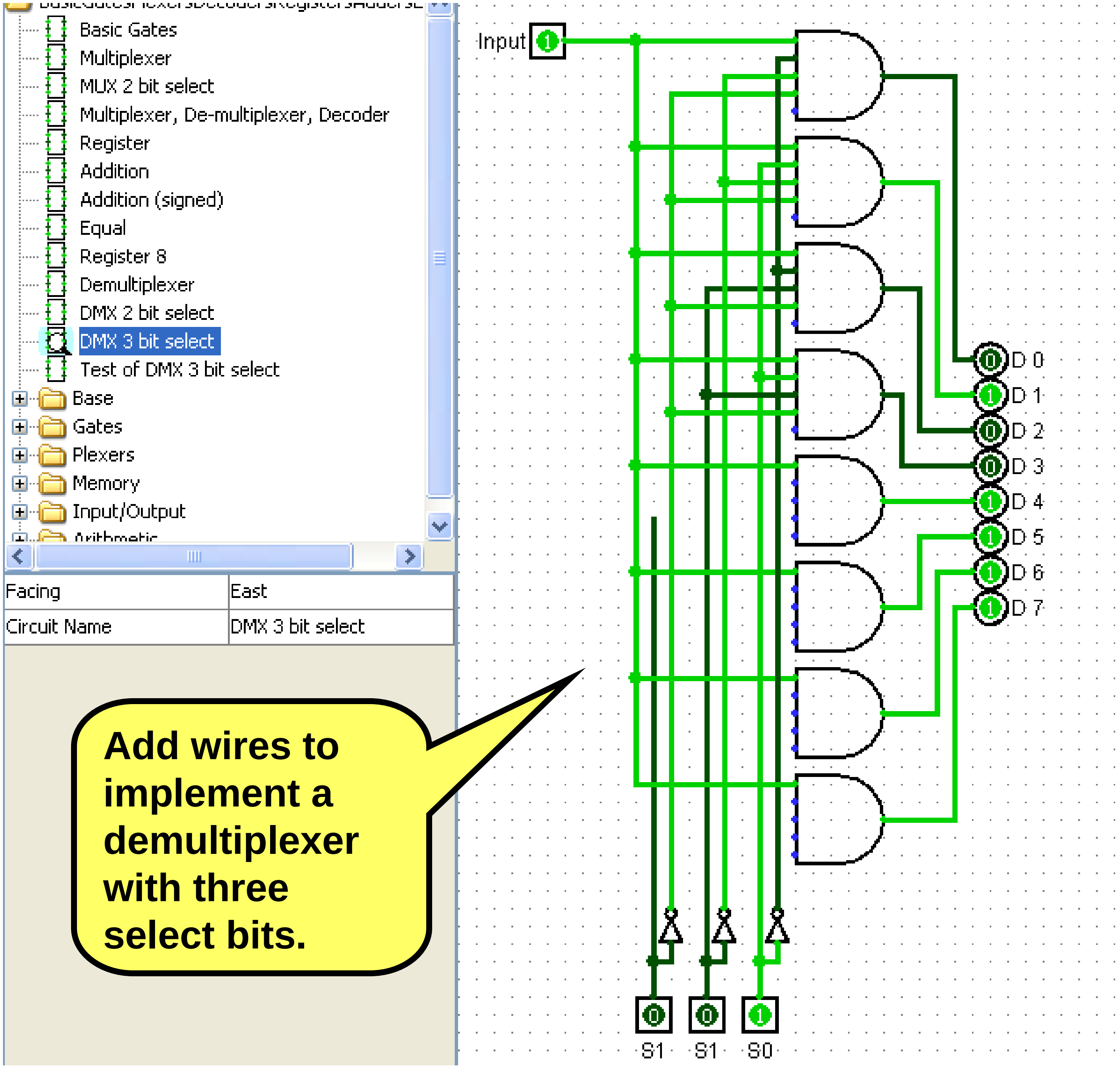
Add wires to implement a demultiplexer with one select bit.



Facing	East
Circuit Name	DMX 2 bit select

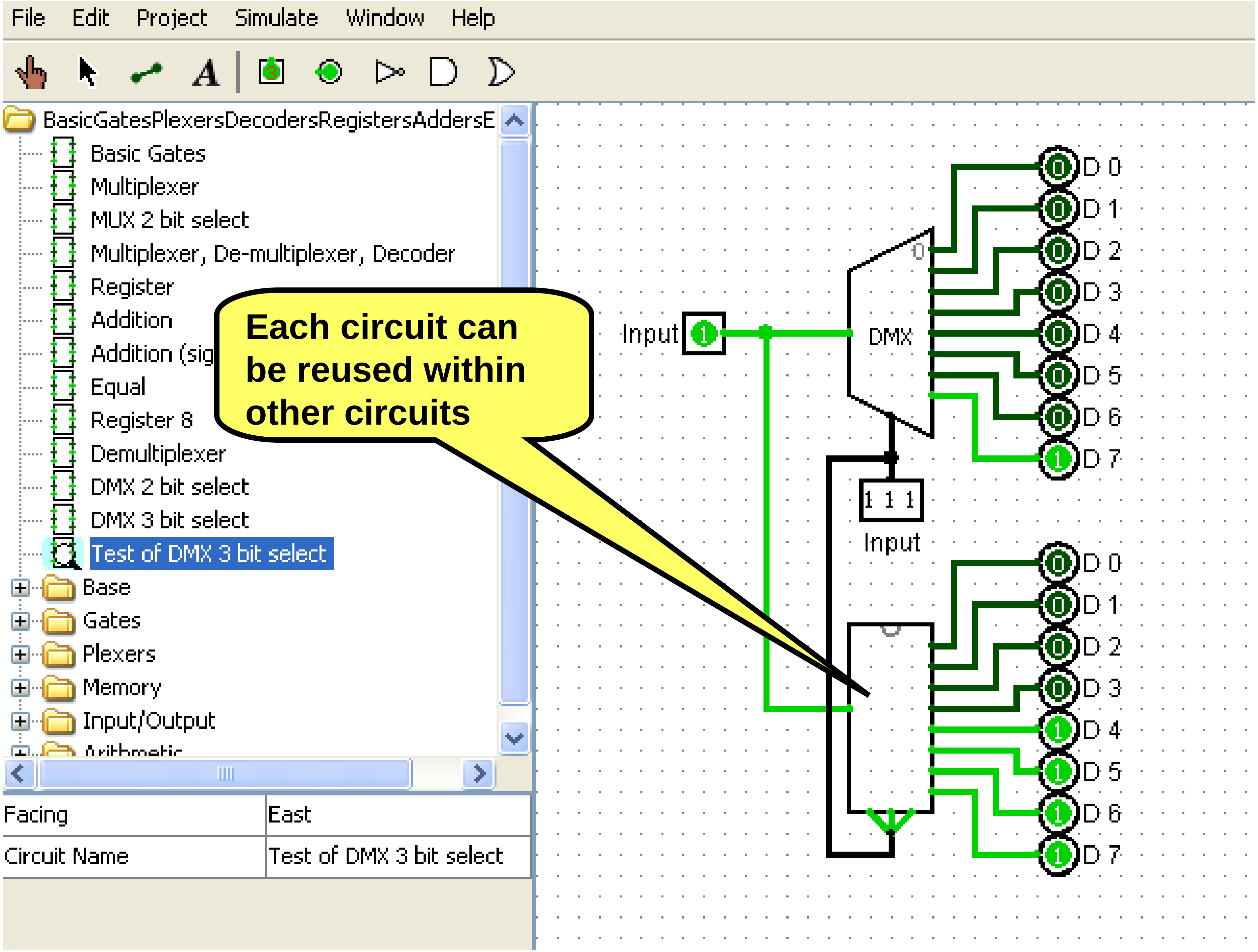
Using two select bits, we can choose to route the input to four different output data lines.

Add wires to implement a demultiplexer with two select bits.



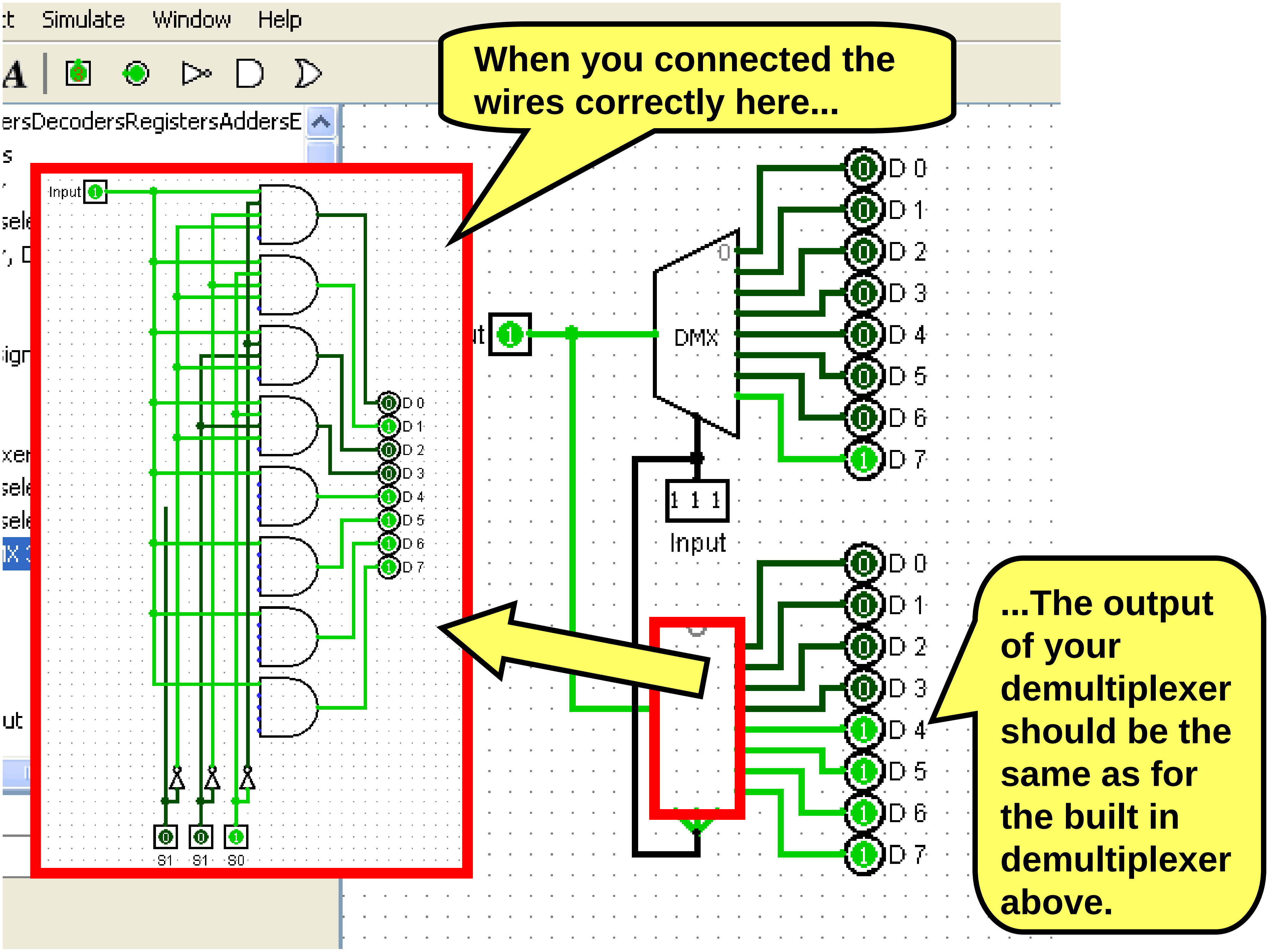
Facing	East
Circuit Name	DMX 3 bit select

Add wires to implement a demultiplexer with three select bits.



Facing	East
Circuit Name	Test of DMX 3 bit select

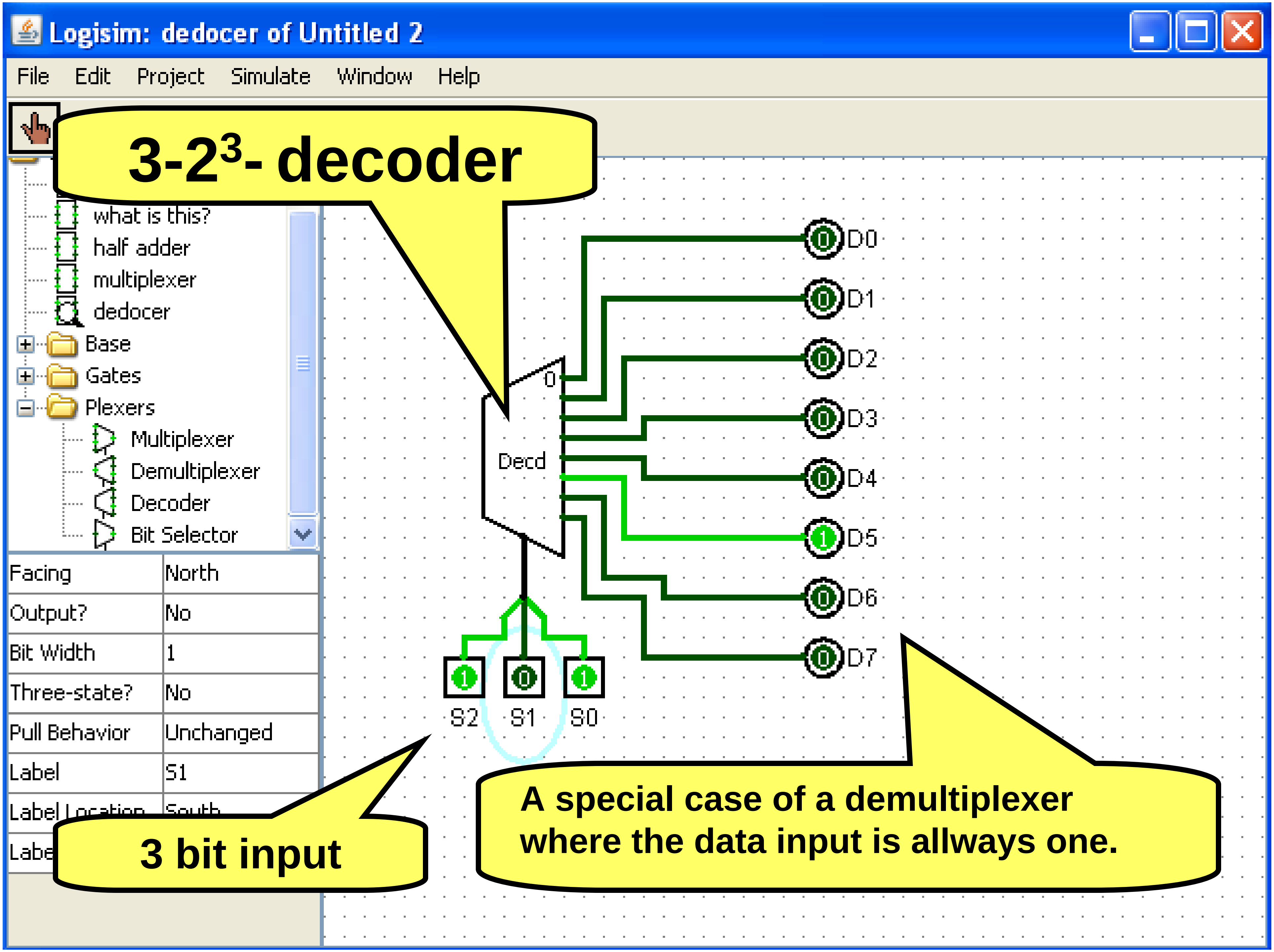
Each circuit can be reused within other circuits



When you connected the wires correctly here...

...The output of your demultiplexer should be the same as for the built in demultiplexer above.

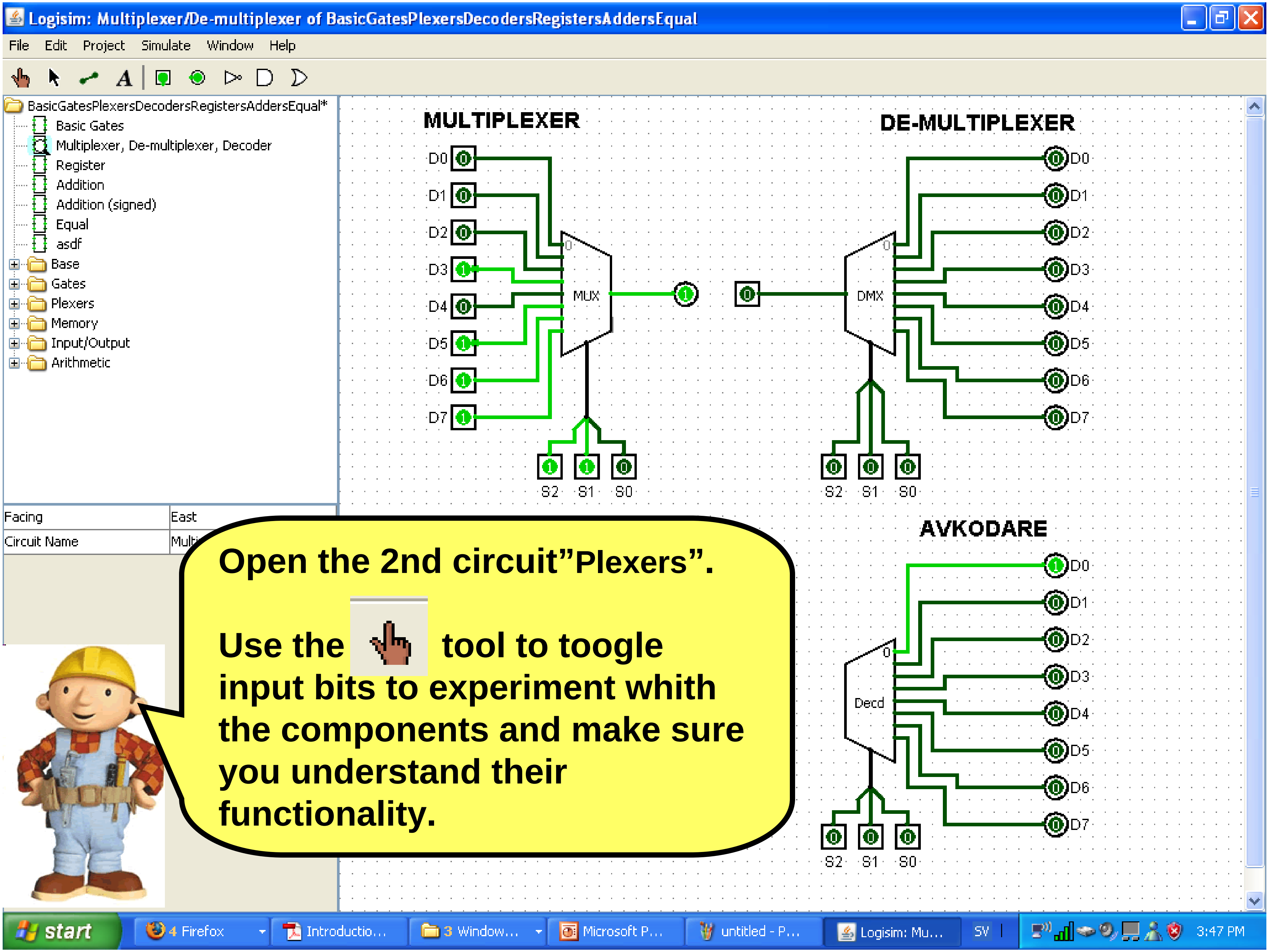
3-2^3- decoder



Facing	North
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	S1
Label Location	South
Labe	

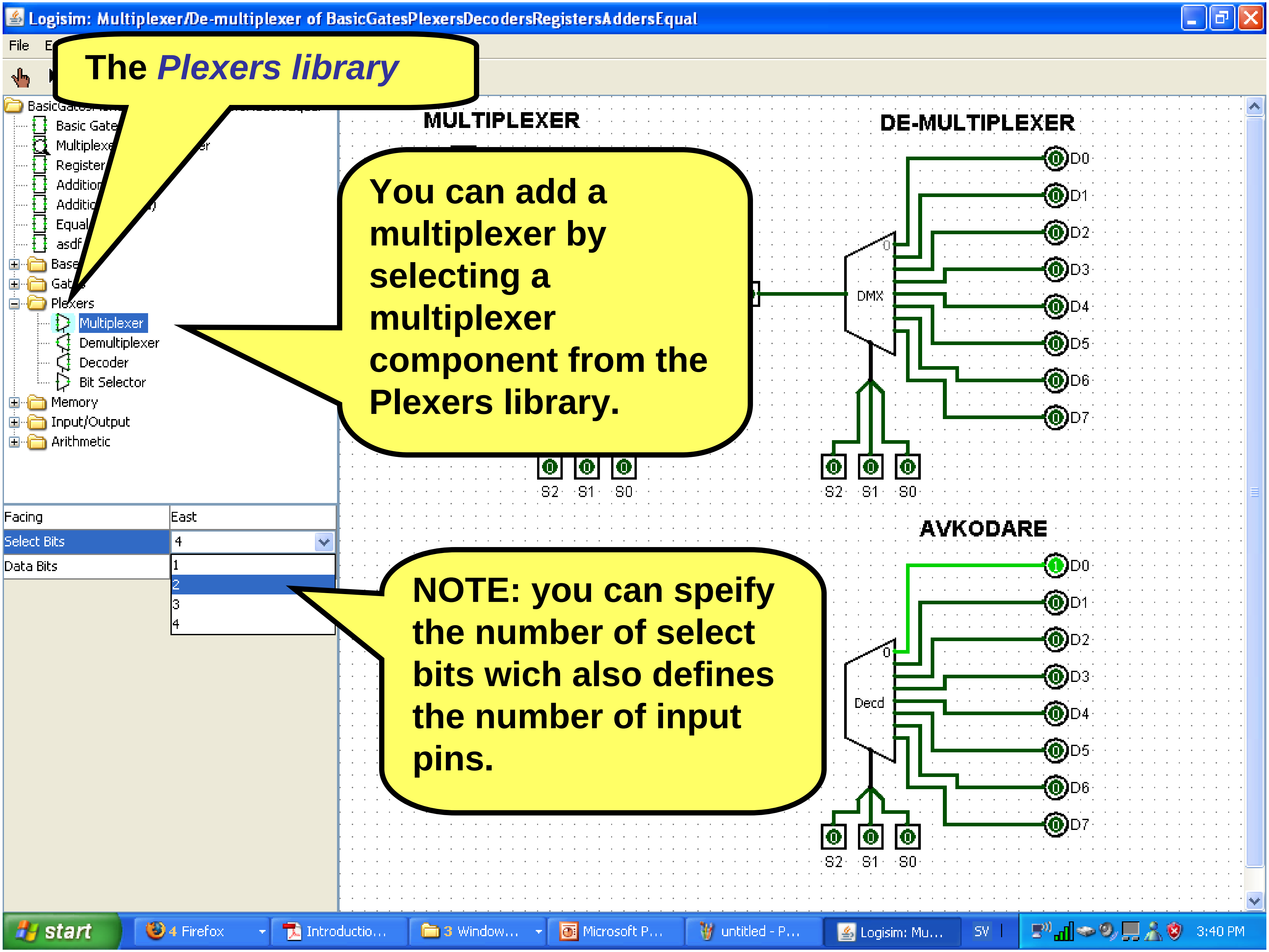
3 bit input

A special case of a demultiplexer where the data input is allways one.



Open the 2nd circuit”Plexers”.

Use the tool to toogle input bits to experiment whith the components and make sure you understand their functionality.



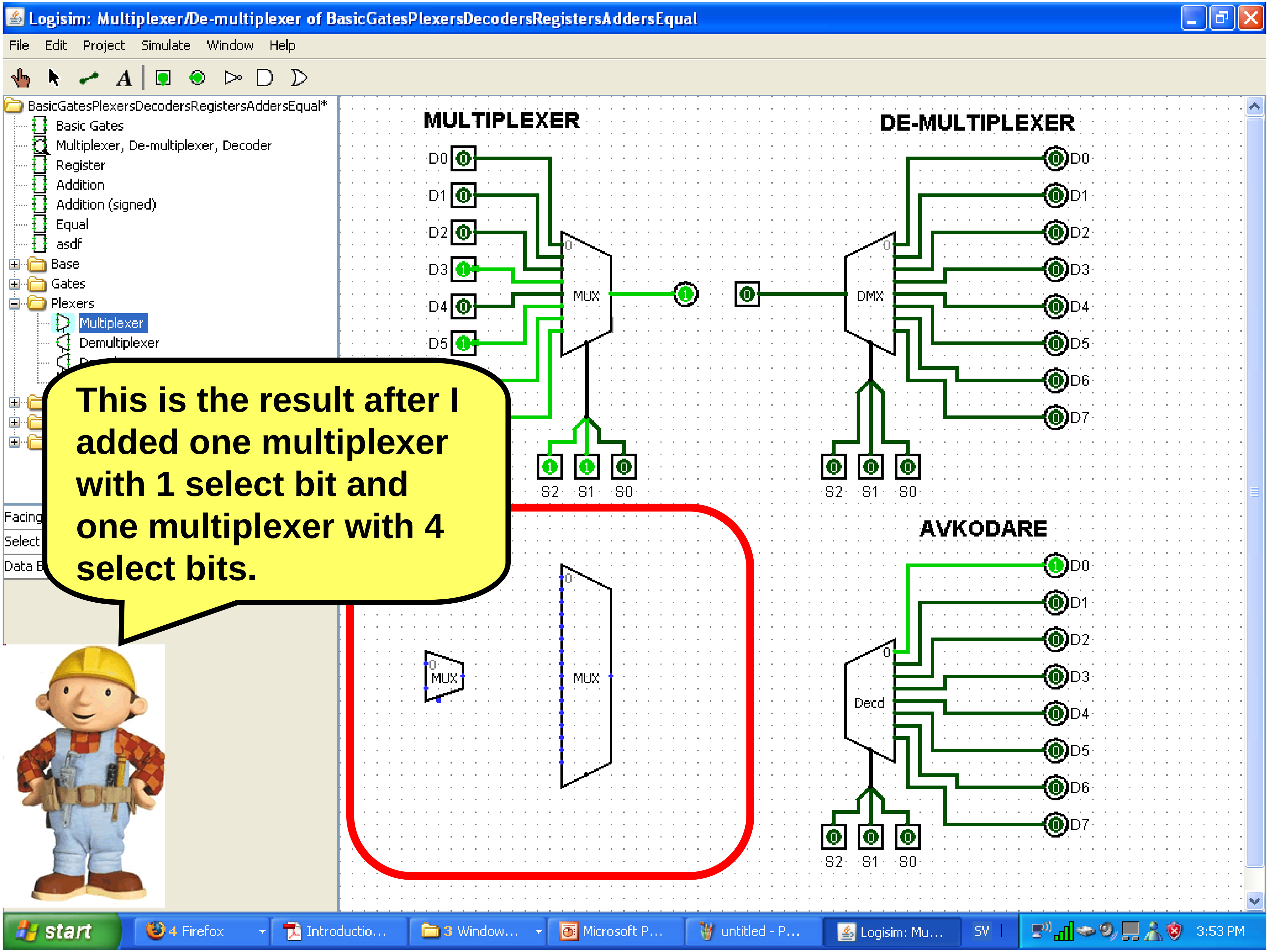
Logisim: Multiplexer/De-multiplexer of BasicGatesPlexersDecodersRegistersAddersEqual

The *Plexers library*

You can add a multiplexer by selecting a multiplexer component from the Plexers library.

Facing	East
Select Bits	4
Data Bits	

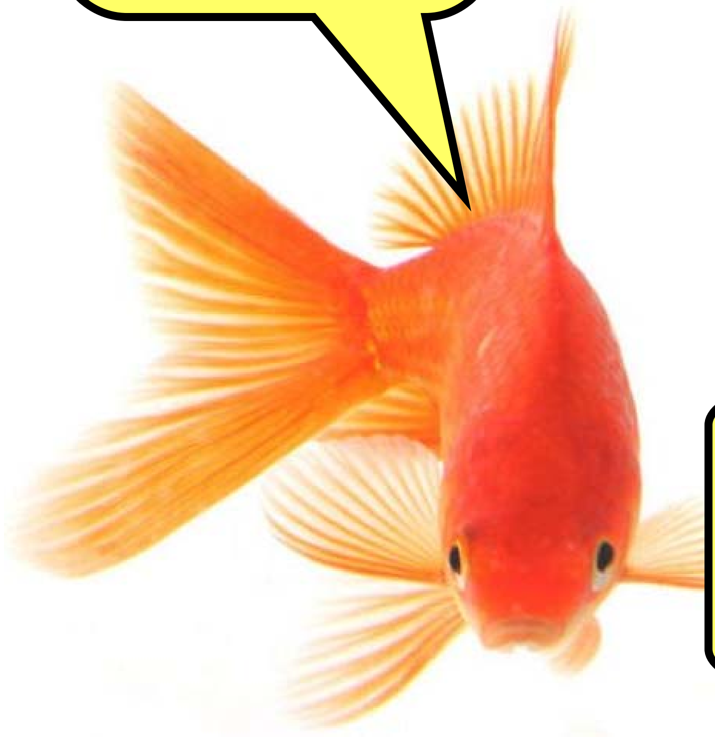
NOTE: you can speify the number of select bits wich also defines the number of input pins.

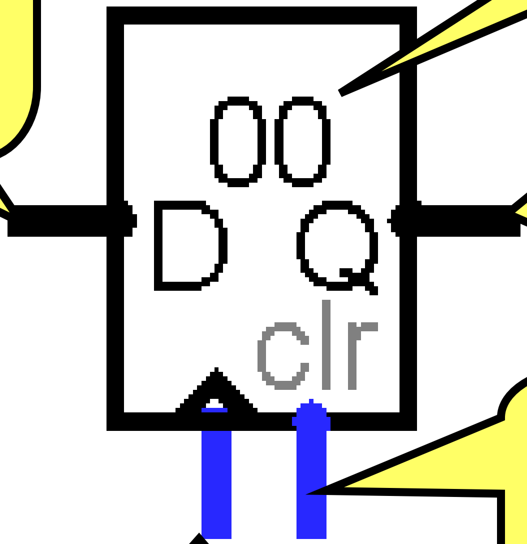


This is the result after I added one multiplexer with 1 select bit and one multiplexer with 4 select bits.

A *register*

We continue by studing a component that can be used to store *integers* (bit patterns).





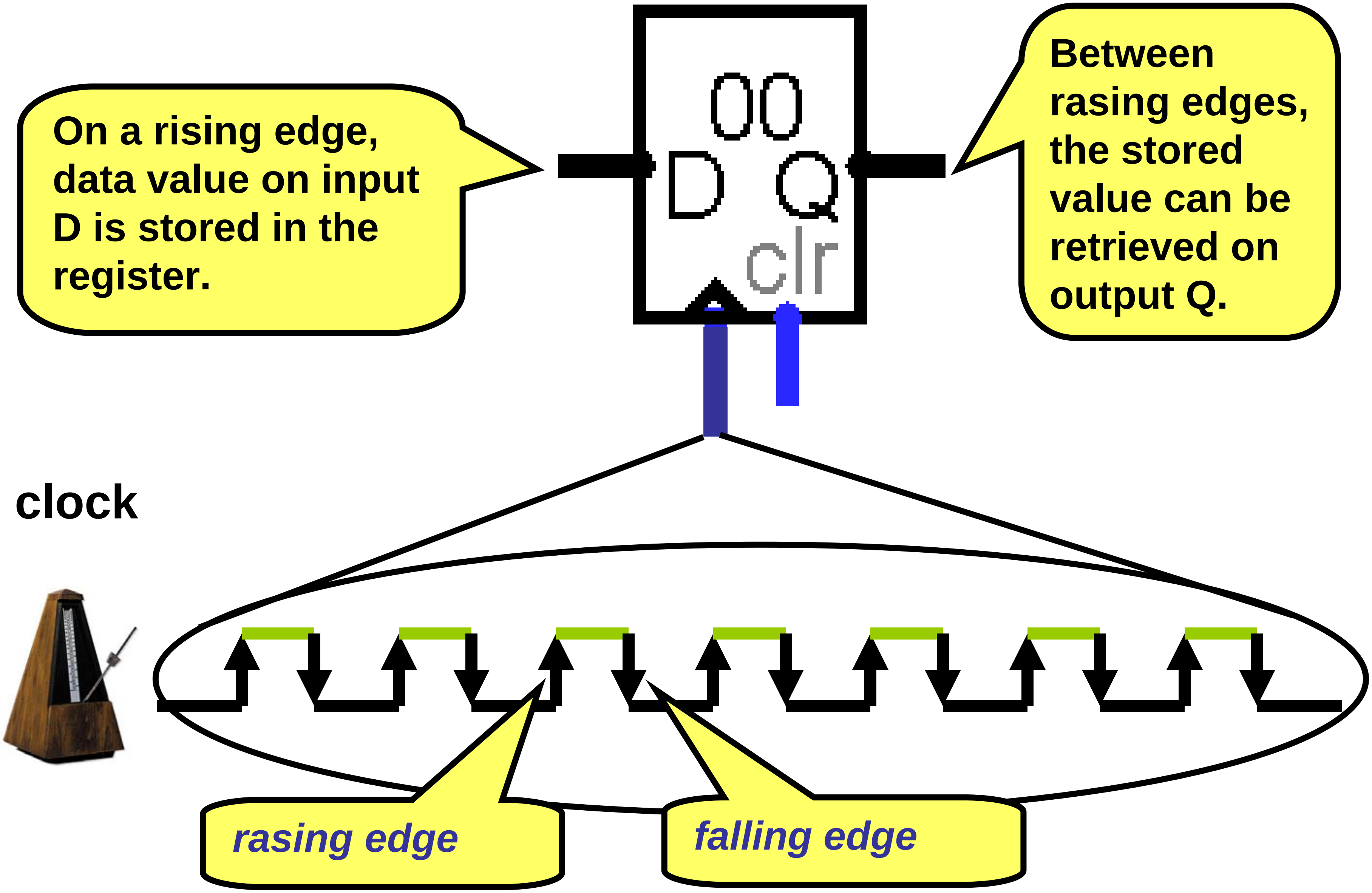
8 bit data input, a number we want to store and *remember*.

Value of stored number in hexadecimal notation.

8 bit data output, value of stored number.

1 bit clear input used to reset stored value to zero.

To change the state (contents) of the register, the value on this 1 bit input must change from low (0) to high (1), i.e., a *rasing edge trigged state element*.



On a rising edge, data value on input D is stored in the register.

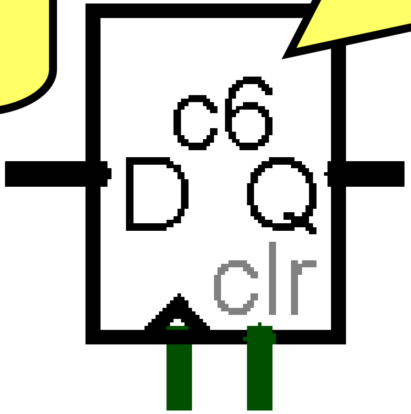
Between rasing edges, the stored value can be retrieved on output Q.

clock

rasing edge

falling edge

What number is stored in this register?

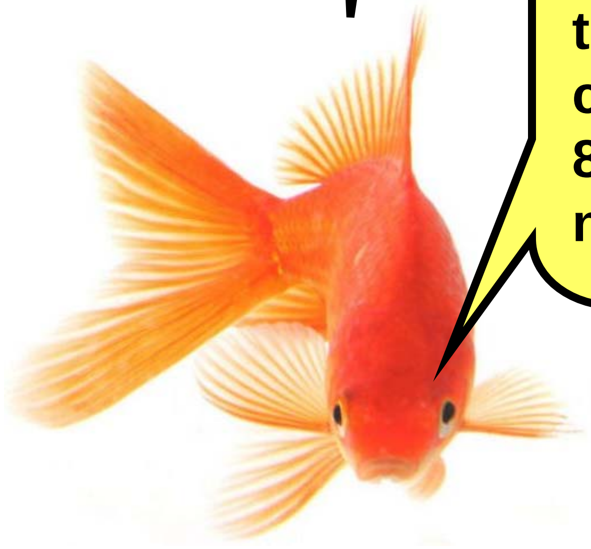


$C6_{16} = 12*16 + 6 = 160 + 32 + 6 = 198_{10}$

TRICK!

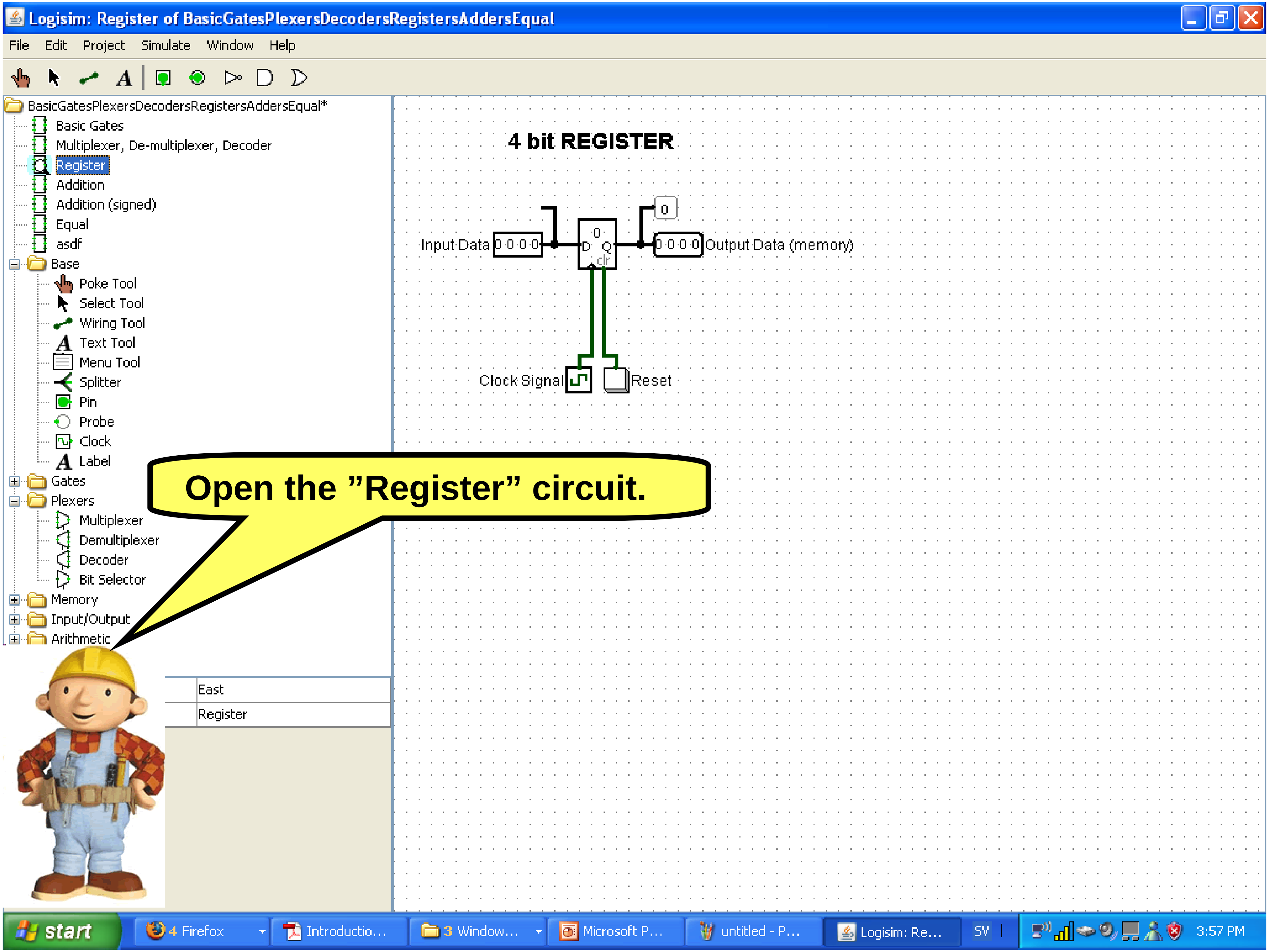
Can tranlate each hexadecimal digit to four binary bits...

Hex	Decimal		Binary
C	12	8 + 4	1100
6	6	4 + 2	0110

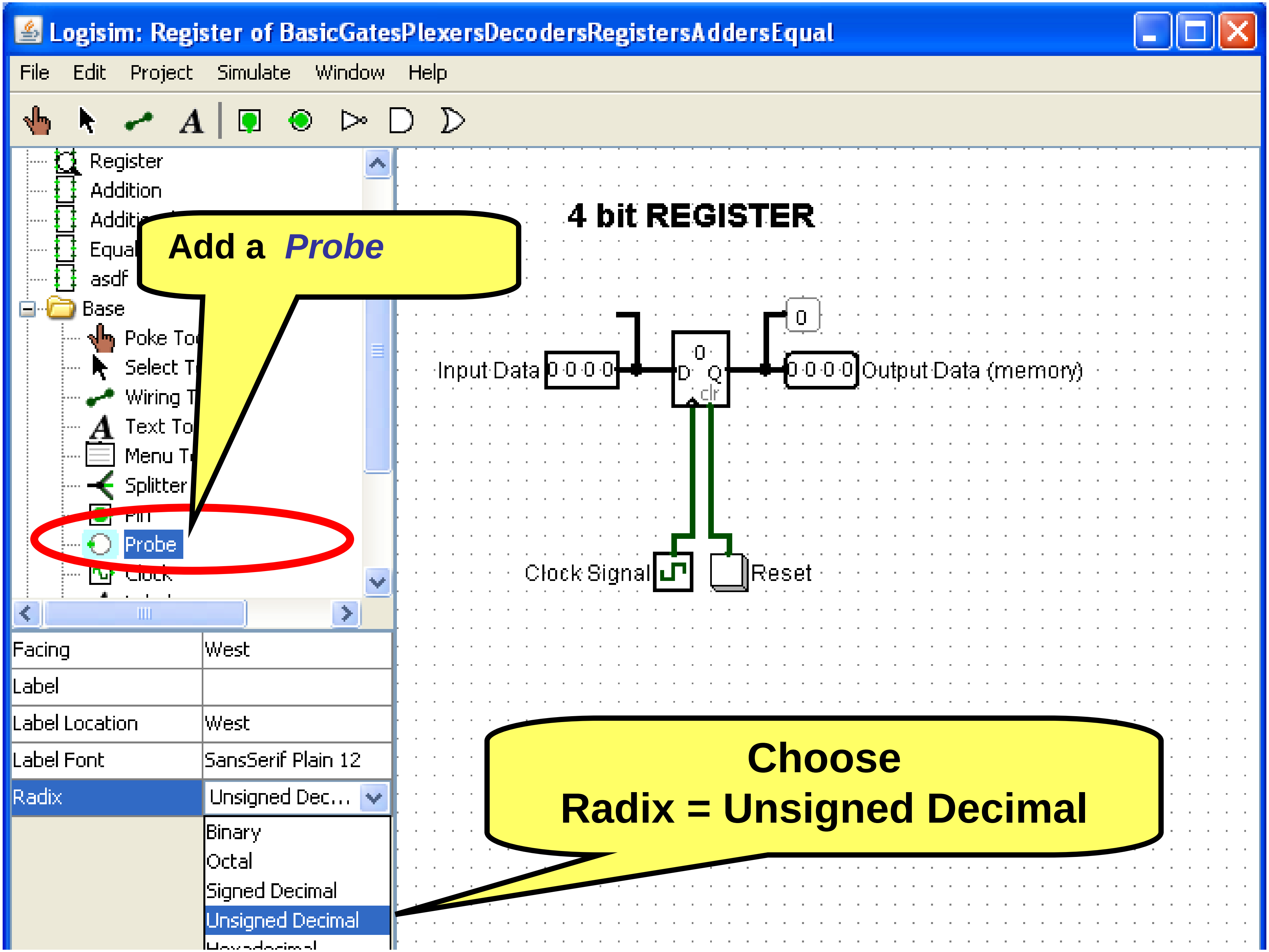


...and assemble to the corresponding 8 bit binay number.

$$C6_{16} = 1100\ 0110_2$$



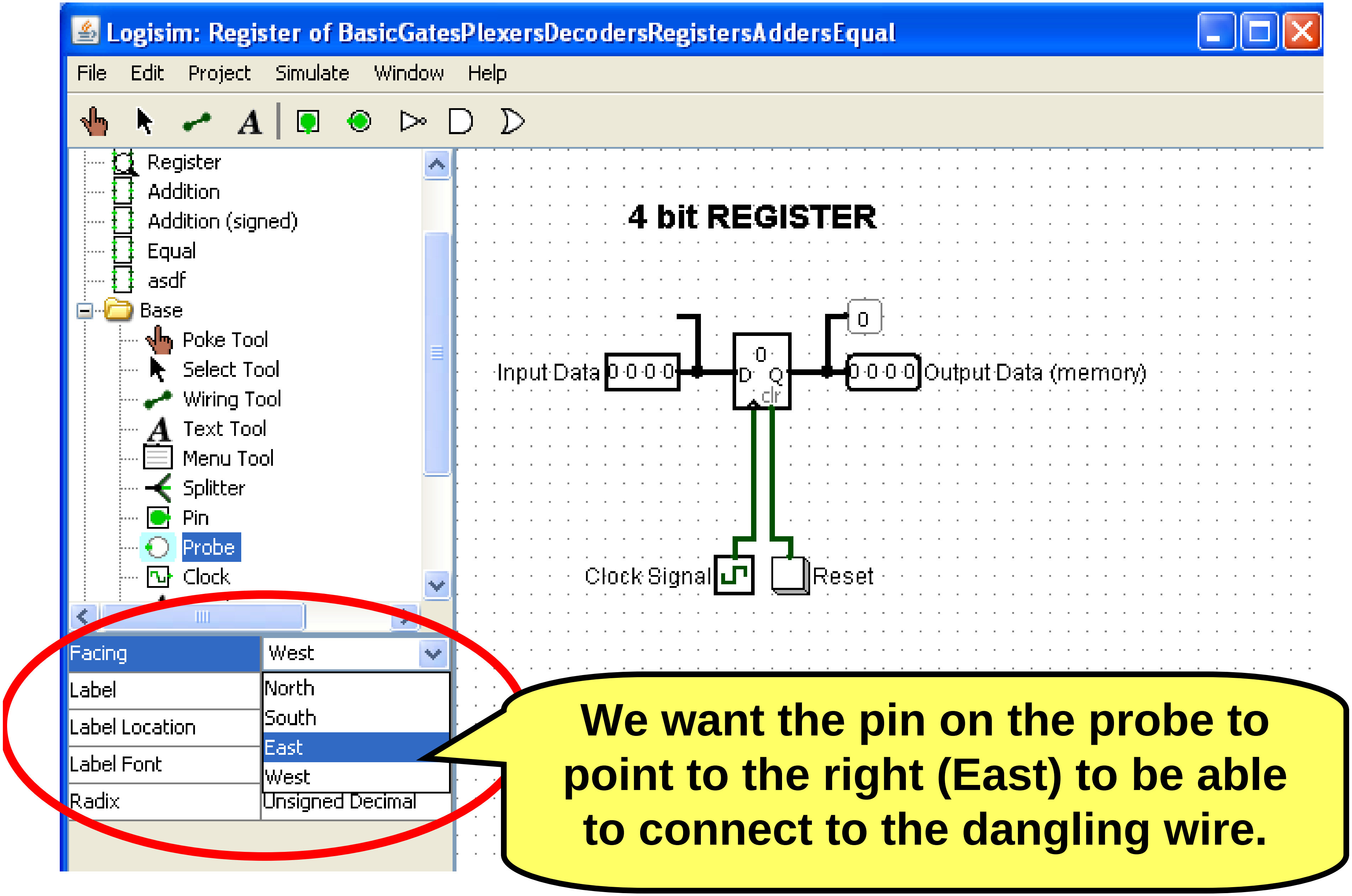
Open the "Register" circuit.



Facing	West
Label	
Label Location	West
Label Font	SansSerif Plain 12
Radix	Unsigned Dec...

Add a *Probe*

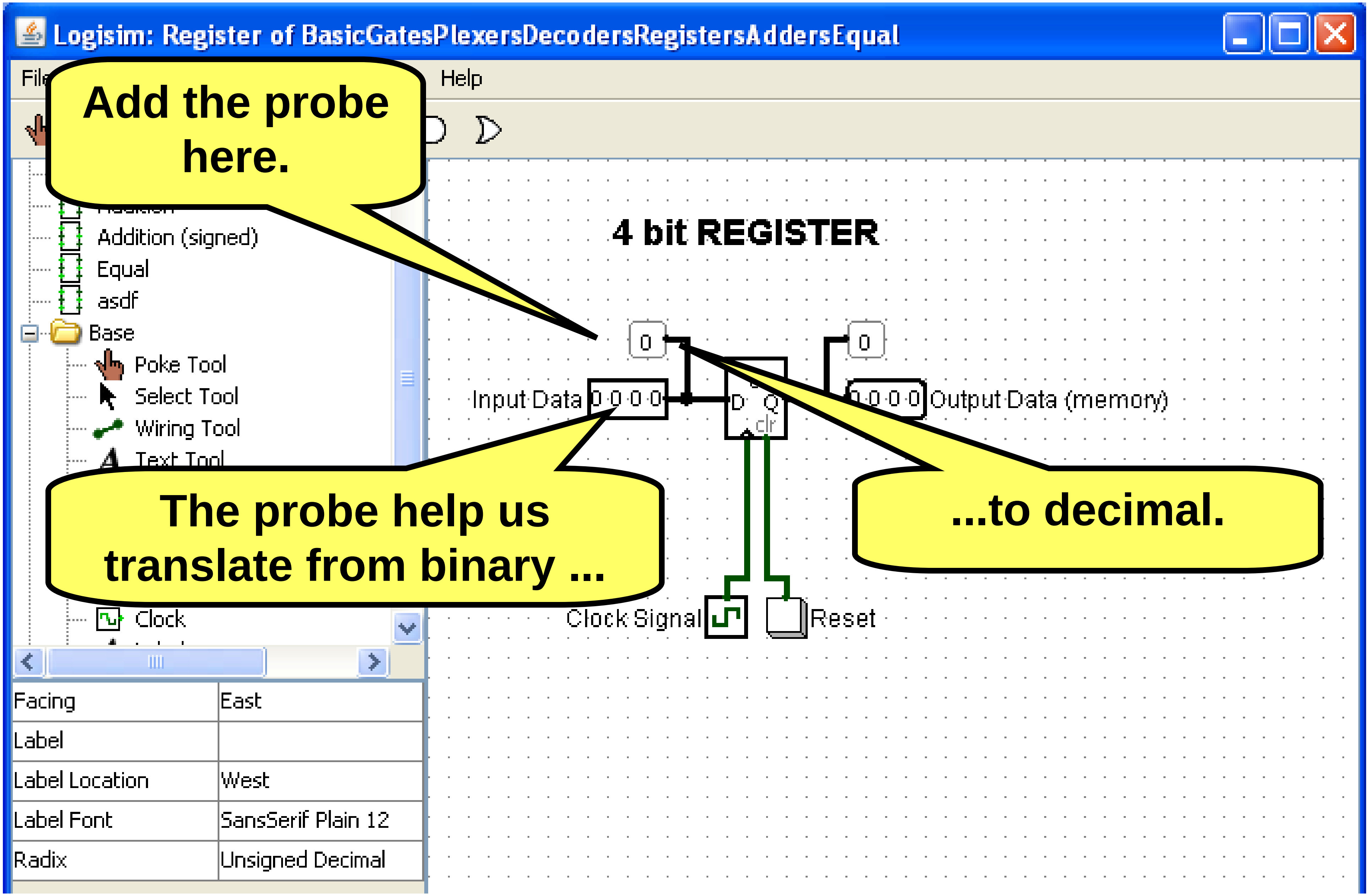
Choose
Radix = Unsigned Decimal



Logisim: Register of BasicGatesPlexersDecodersRegistersAddersEqual

Facing	West
Label	
Label Location	
Label Font	
Radix	Unsigned Decimal

We want the pin on the probe to point to the right (East) to be able to connect to the dangling wire.

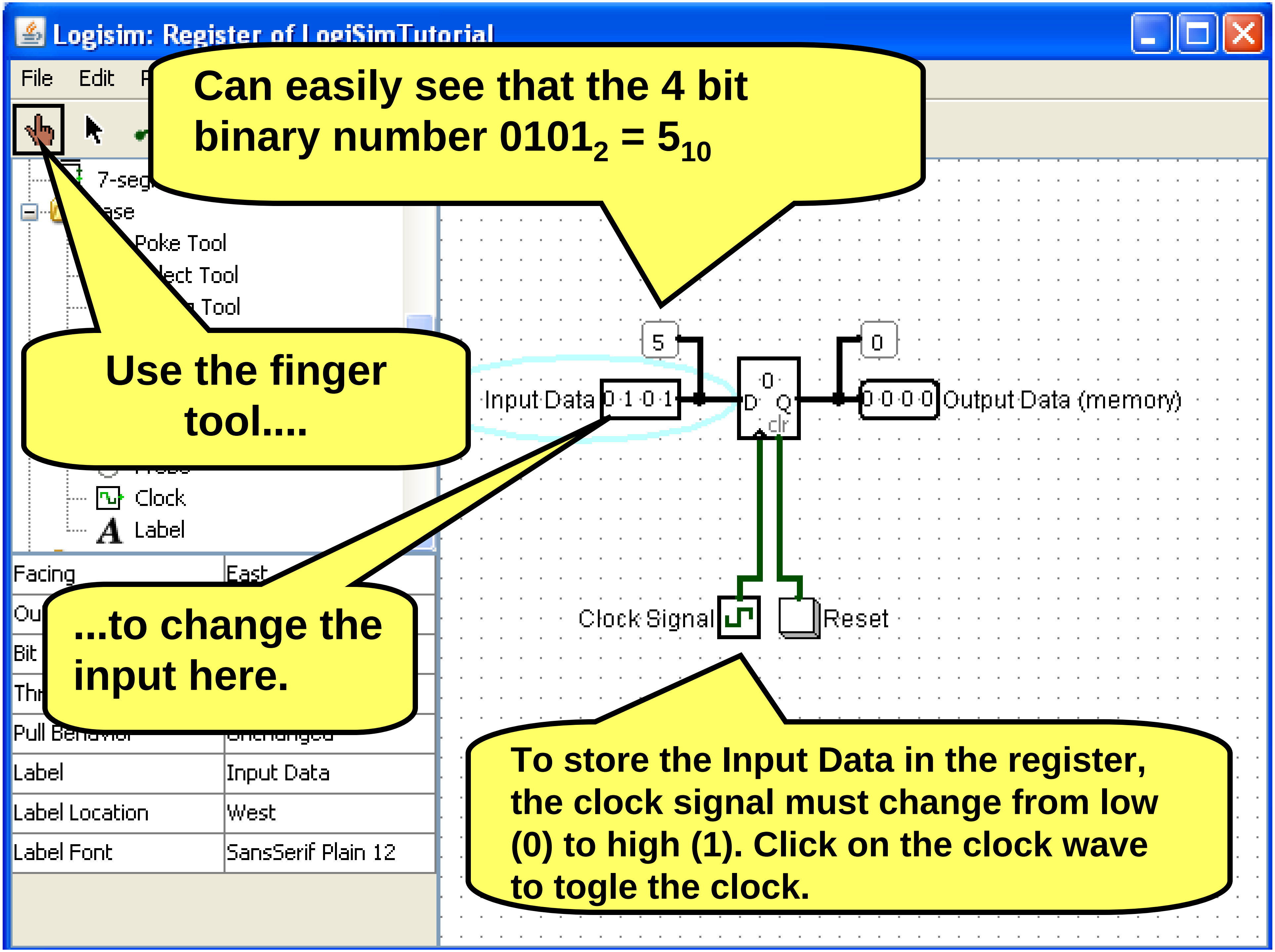


Facing	East
Label	
Label Location	West
Label Font	SansSerif Plain 12
Radix	Unsigned Decimal

Add the probe here.

The probe help us translate from binary ...

...to decimal.



Facing: East

Label: Input Data

Label Location: West

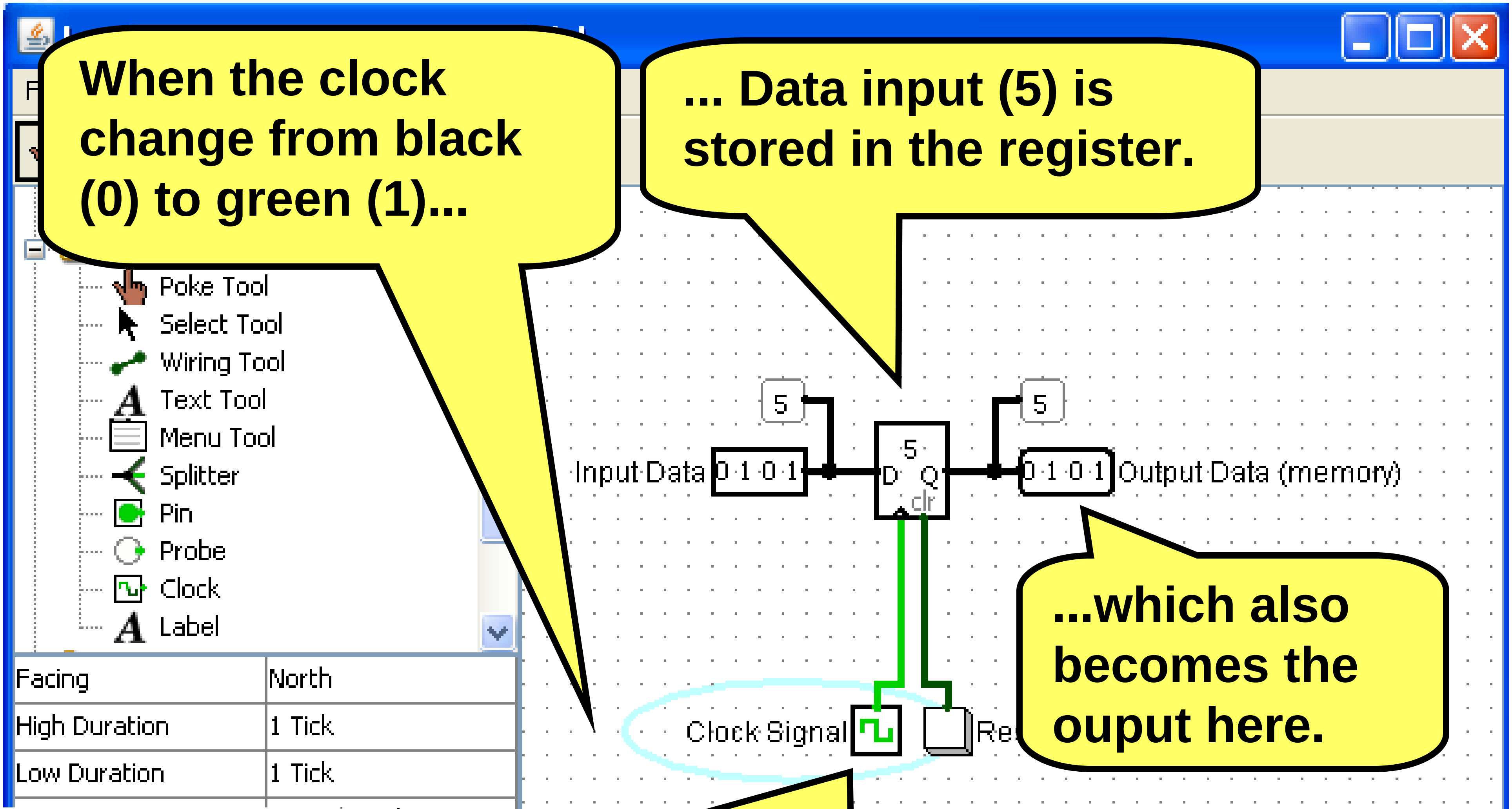
Label Font: SansSerif Plain 12

Can easily see that the 4 bit binary number $0101_2 = 5_{10}$

Use the finger tool....

...to change the input here.

To store the Input Data in the register, the clock signal must change from low (0) to high (1). Click on the clock wave to togle the clock.



Facing	North
High Duration	1 Tick
Low Duration	1 Tick
	ignal
	erif Pla

When the clock change from black (0) to green (1)...

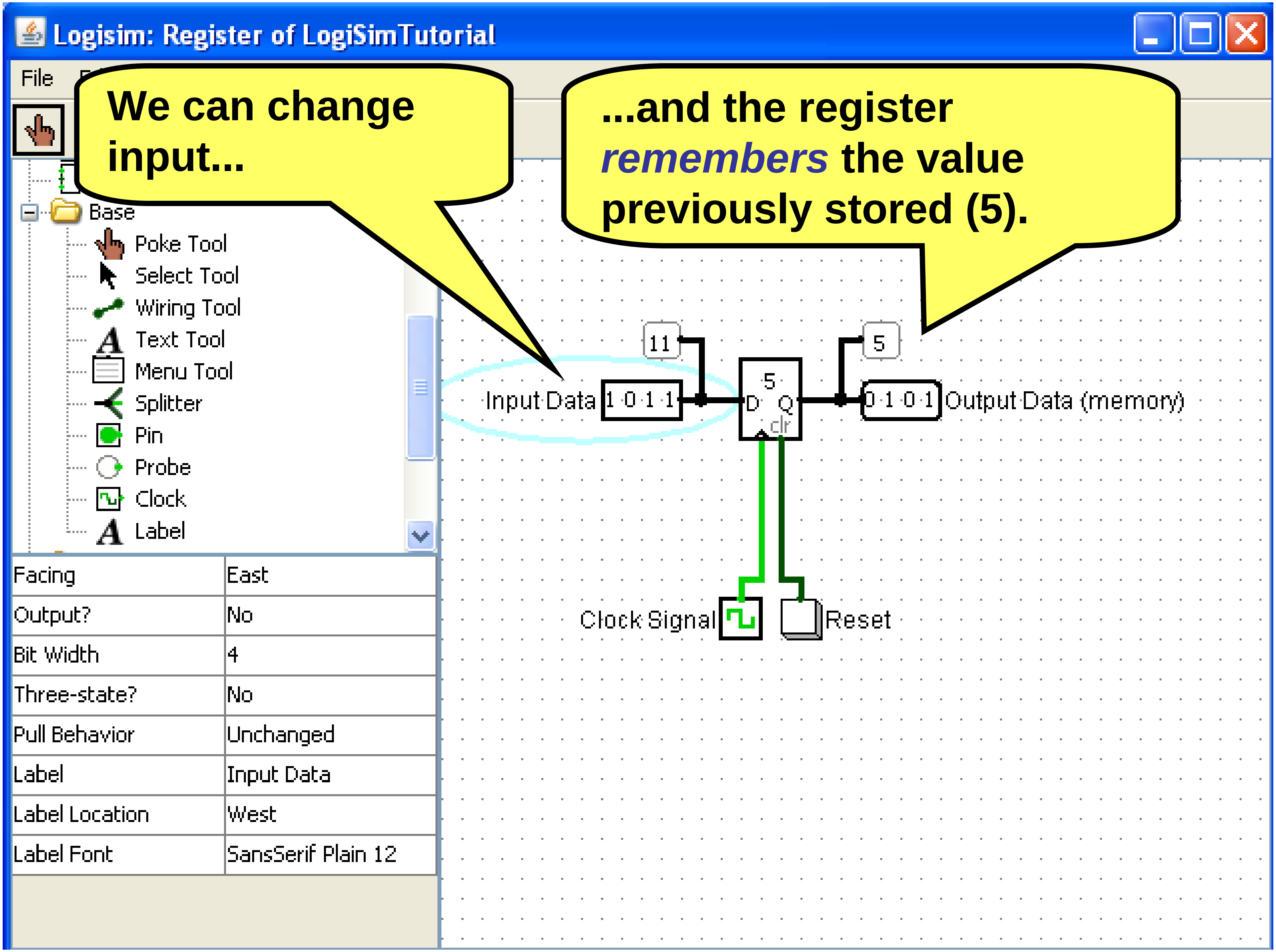
... Data input (5) is stored in the register.

...which also becomes the ouput here.

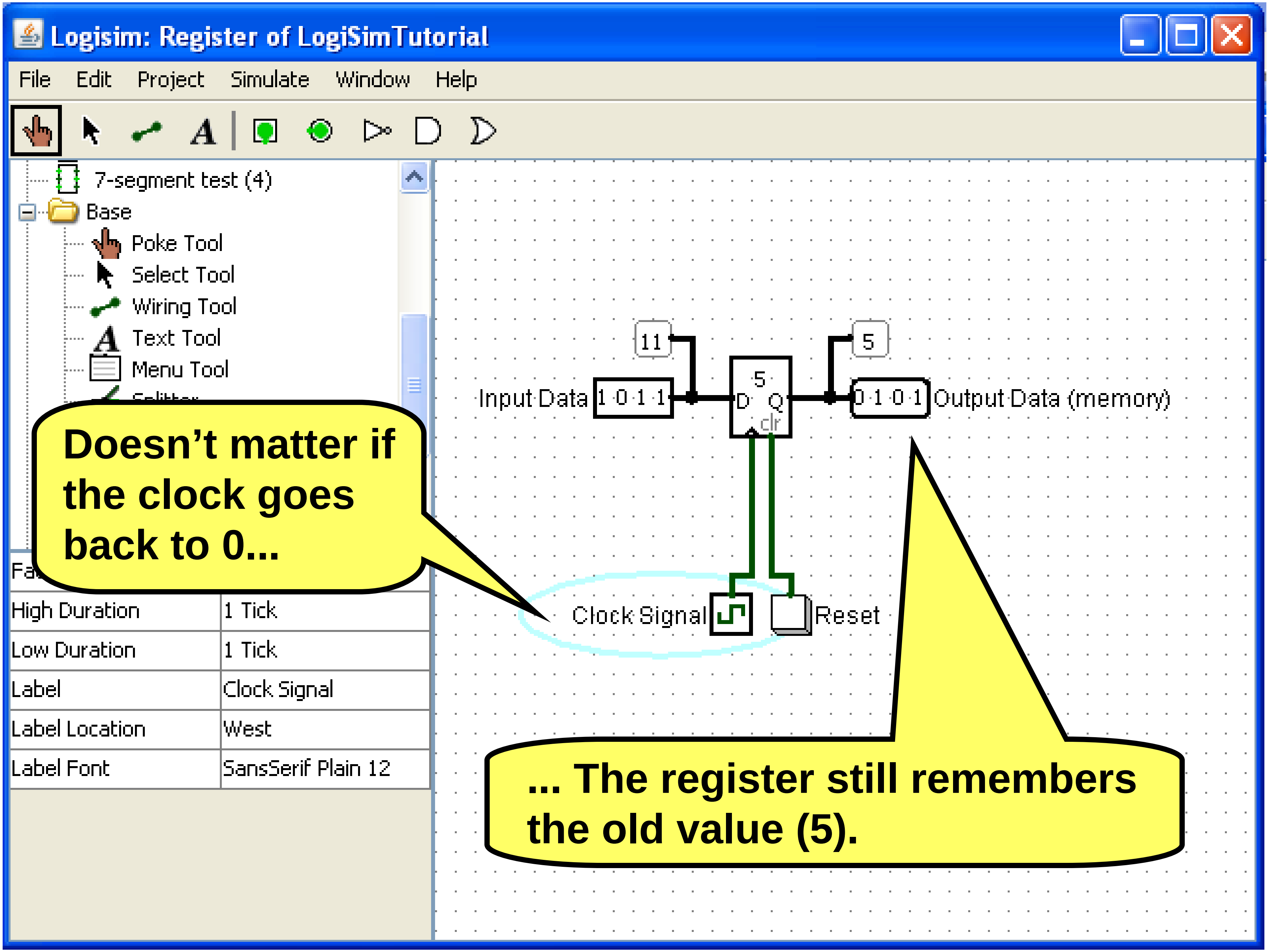
Do you remember: To change the state (contents) of the register, the value of the 1 bit clock signal must change from low (0) to high (1), i.e., a *rasing edge trigged state element*.

We can change input...

...and the register *remembers* the value previously stored (5).



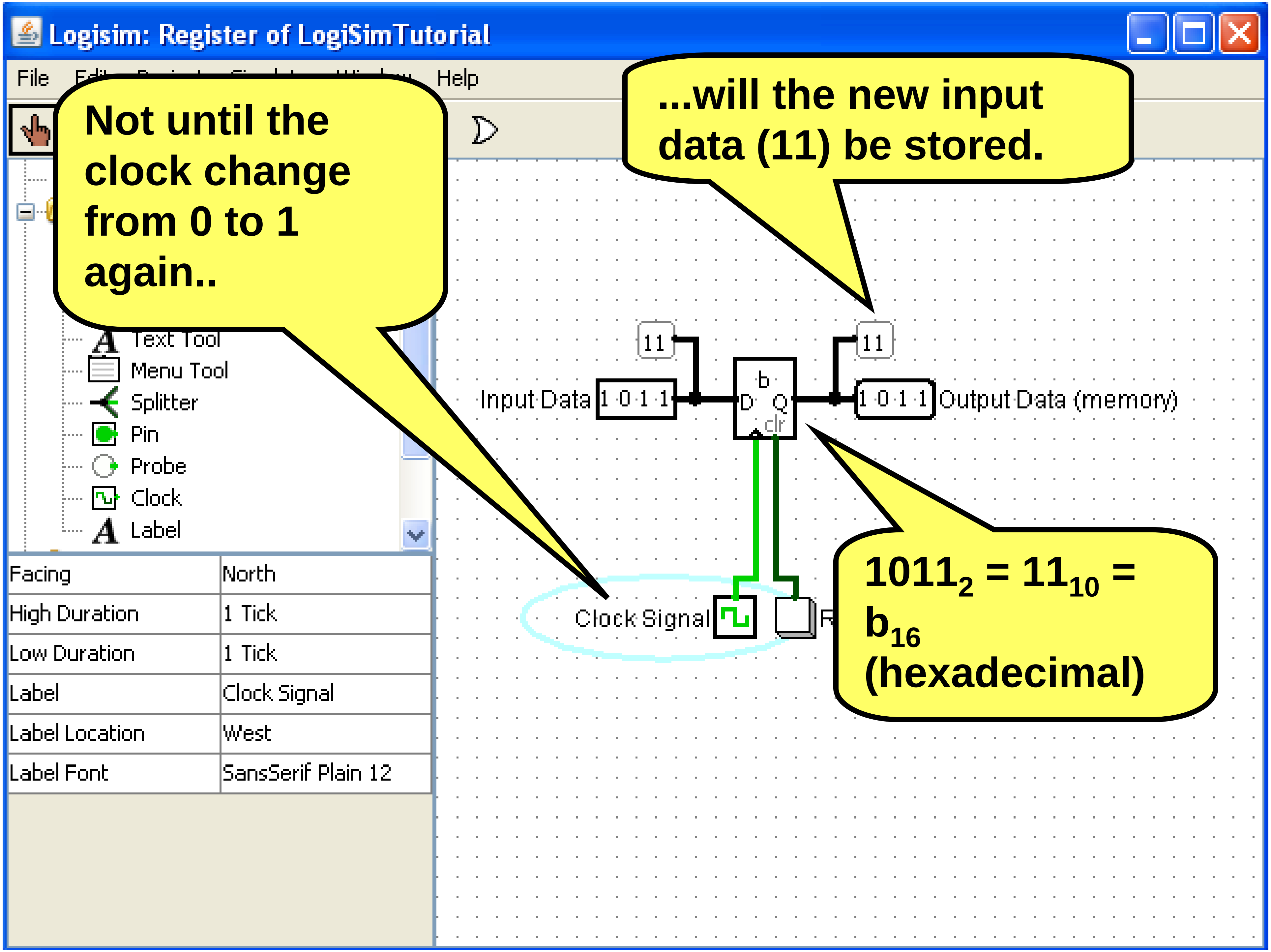
Facing	East
Output?	No
Bit Width	4
Three-state?	No
Pull Behavior	Unchanged
Label	Input Data
Label Location	West
Label Font	SansSerif Plain 12



Fa	
High Duration	1 Tick
Low Duration	1 Tick
Label	Clock Signal
Label Location	West
Label Font	SansSerif Plain 12

Doesn't matter if the clock goes back to 0...

... The register still remembers the old value (5).

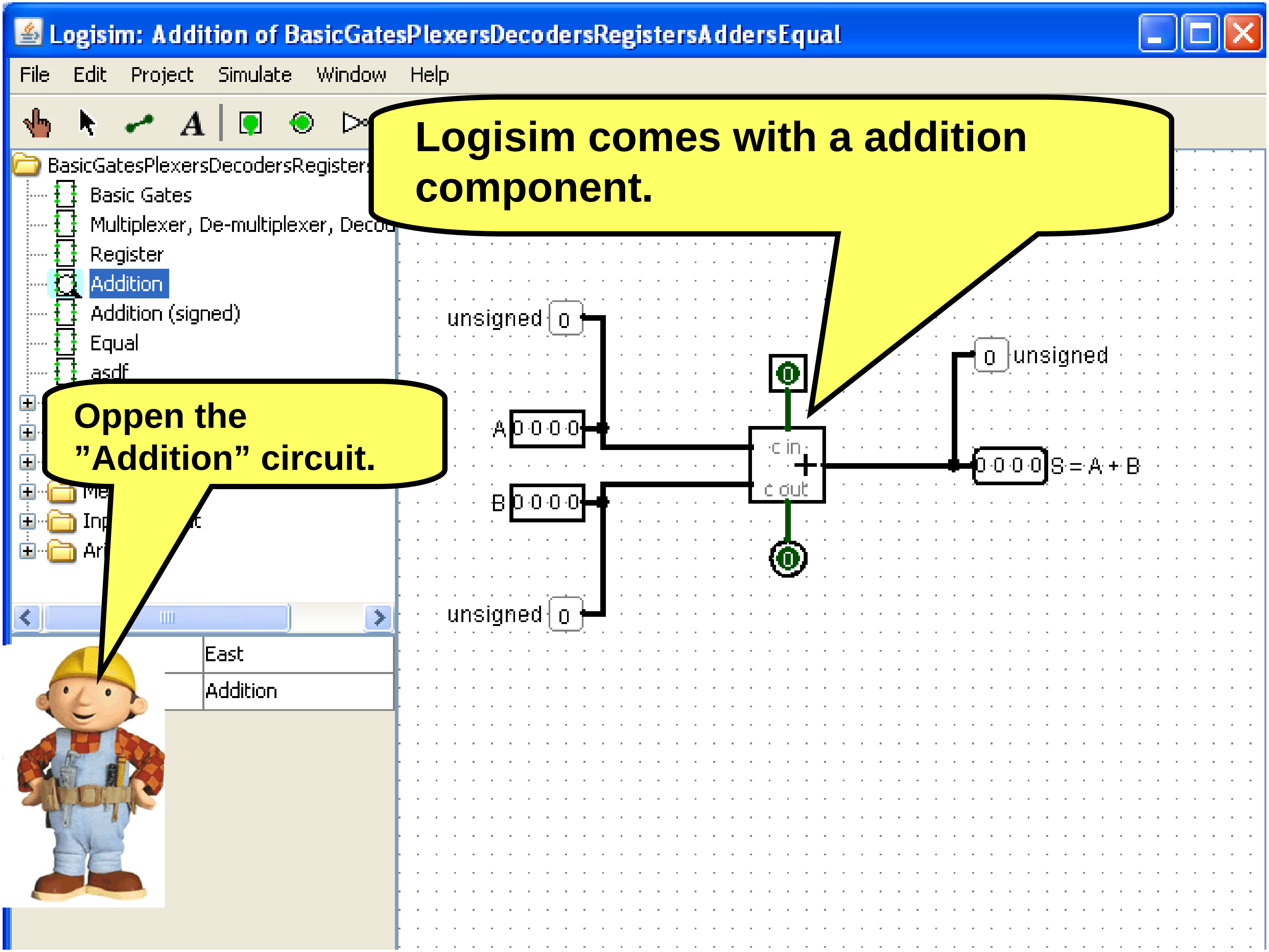


Facing	North
High Duration	1 Tick
Low Duration	1 Tick
Label	Clock Signal
Label Location	West
Label Font	SansSerif Plain 12

Not until the clock change from 0 to 1 again..

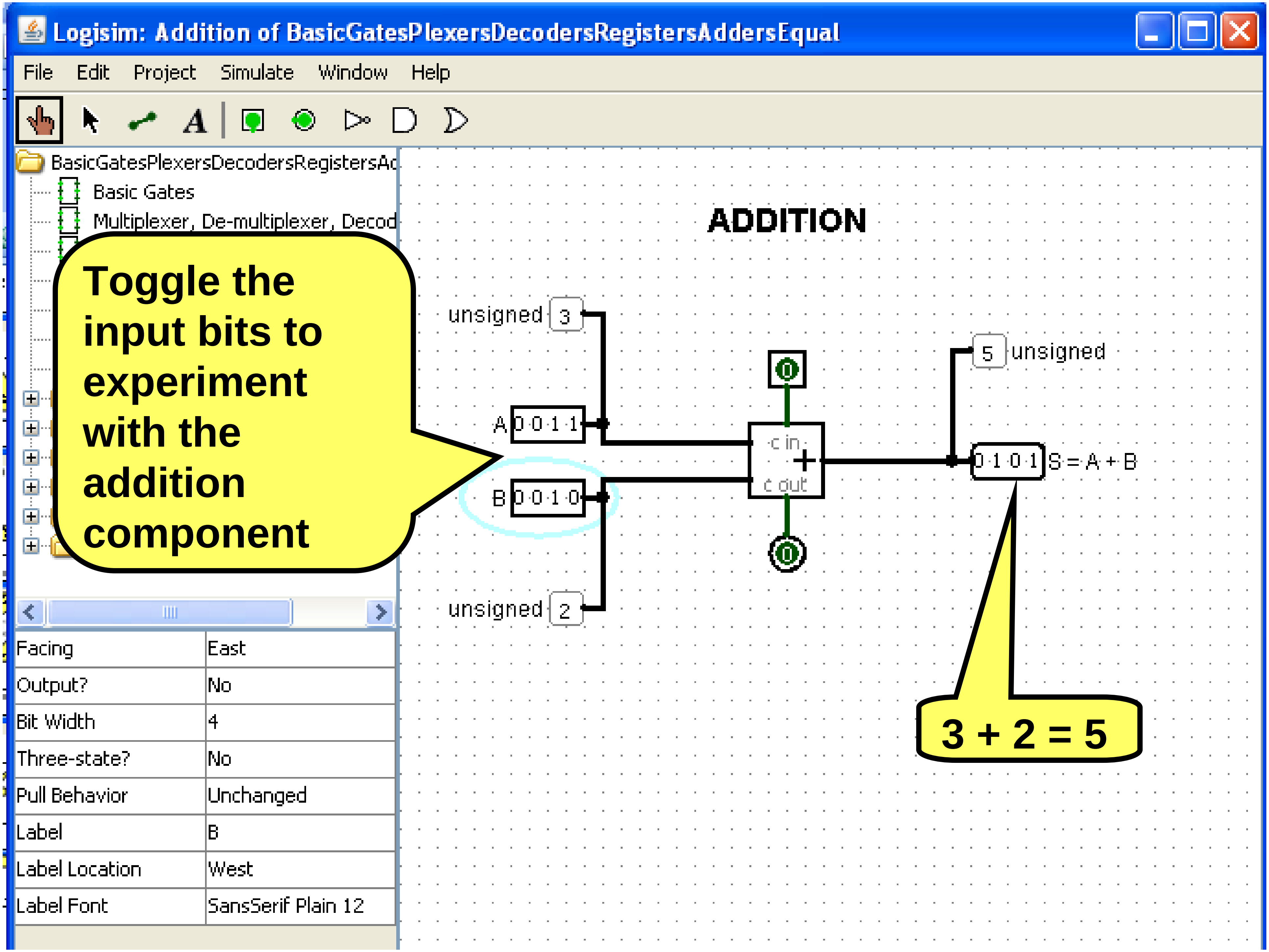
...will the new input data (11) be stored.

$1011_2 = 11_{10} = b_{16}$ (hexadecimal)



Logisim comes with a addition component.

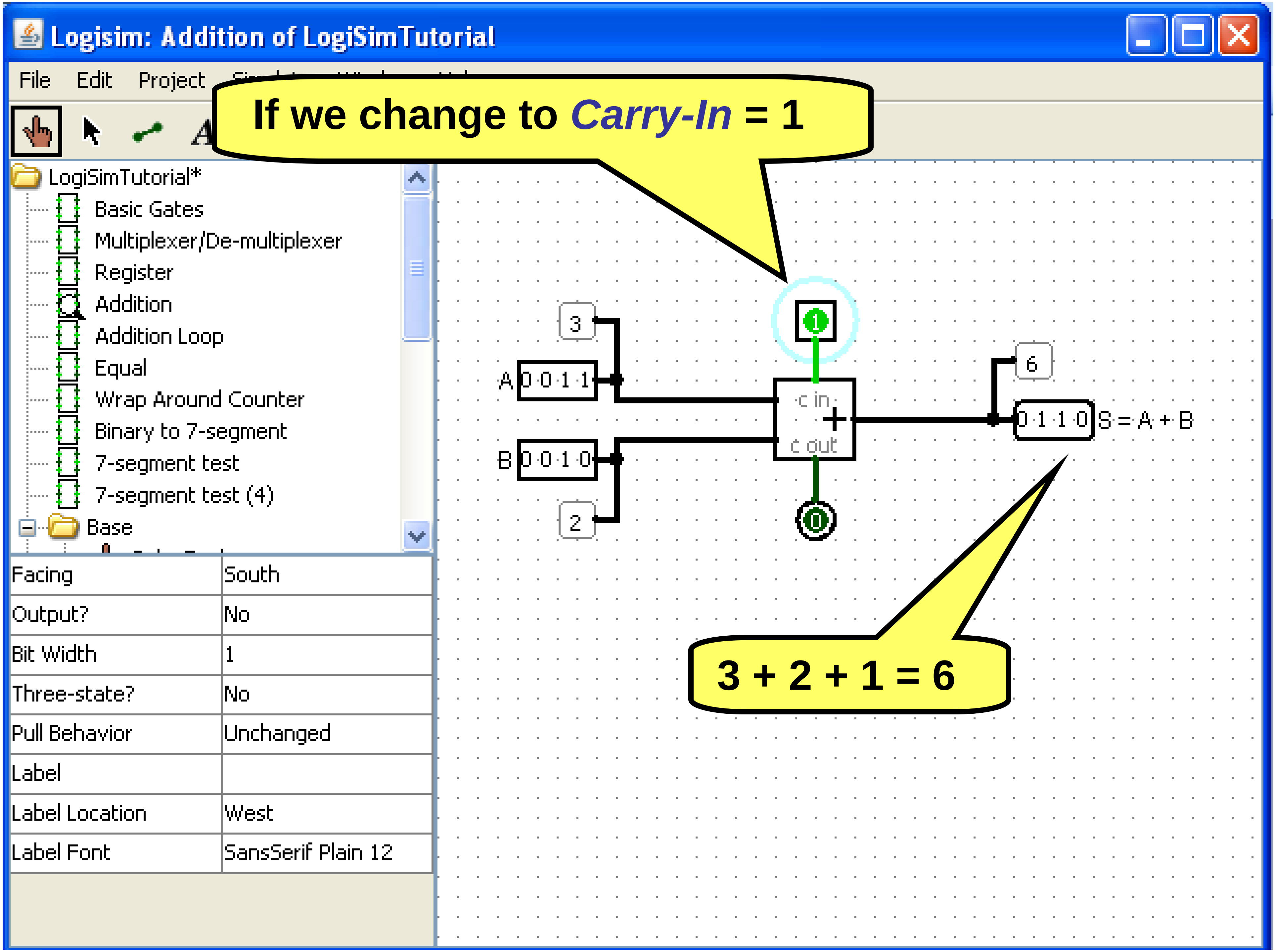
Oppen the ”Addition” circuit.


Logisim: Addition of BasicGatesPlexersDecodersRegistersAddersEqual
File
Edit
Project
Simulate
Window
Help
BasicGatesPlexersDecodersRegistersAc
Basic Gates
Multiplexer, De-multiplexer, Decod
ADDITION
unsigned 3
A 0 0 1 1
B 0 0 1 0
unsigned 2
c in
+
c out
0
0
0 1 0 1 S = A + B
5 unsigned

Facing	East
Output?	No
Bit Width	4
Three-state?	No
Pull Behavior	Unchanged
Label	B
Label Location	West
Label Font	SansSerif Plain 12

Toggle the input bits to experiment with the addition component

3 + 2 = 5

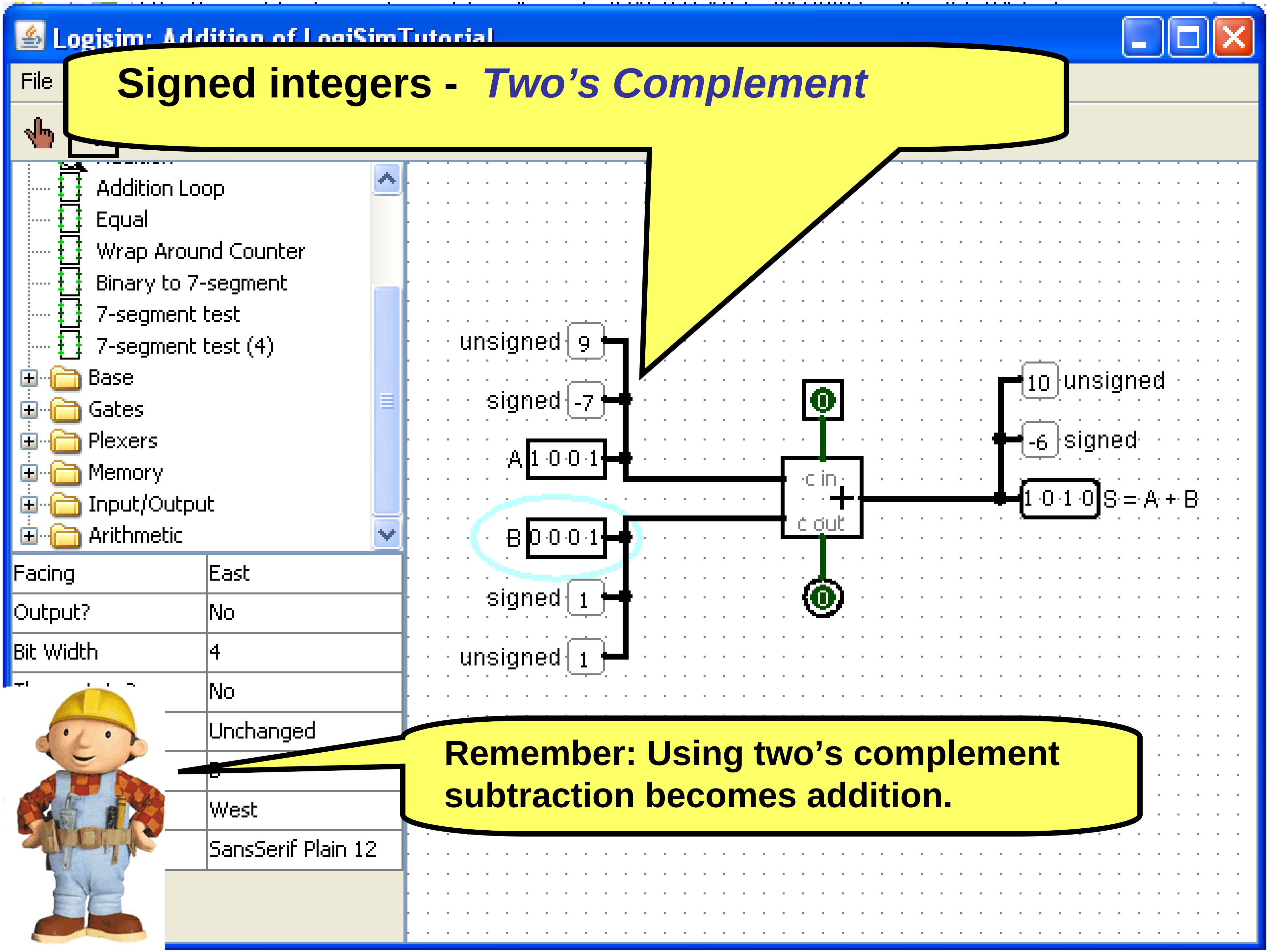


If we change to *Carry-In* = 1

3 + 2 + 1 = 6

Facing	South
Output?	No
Bit Width	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

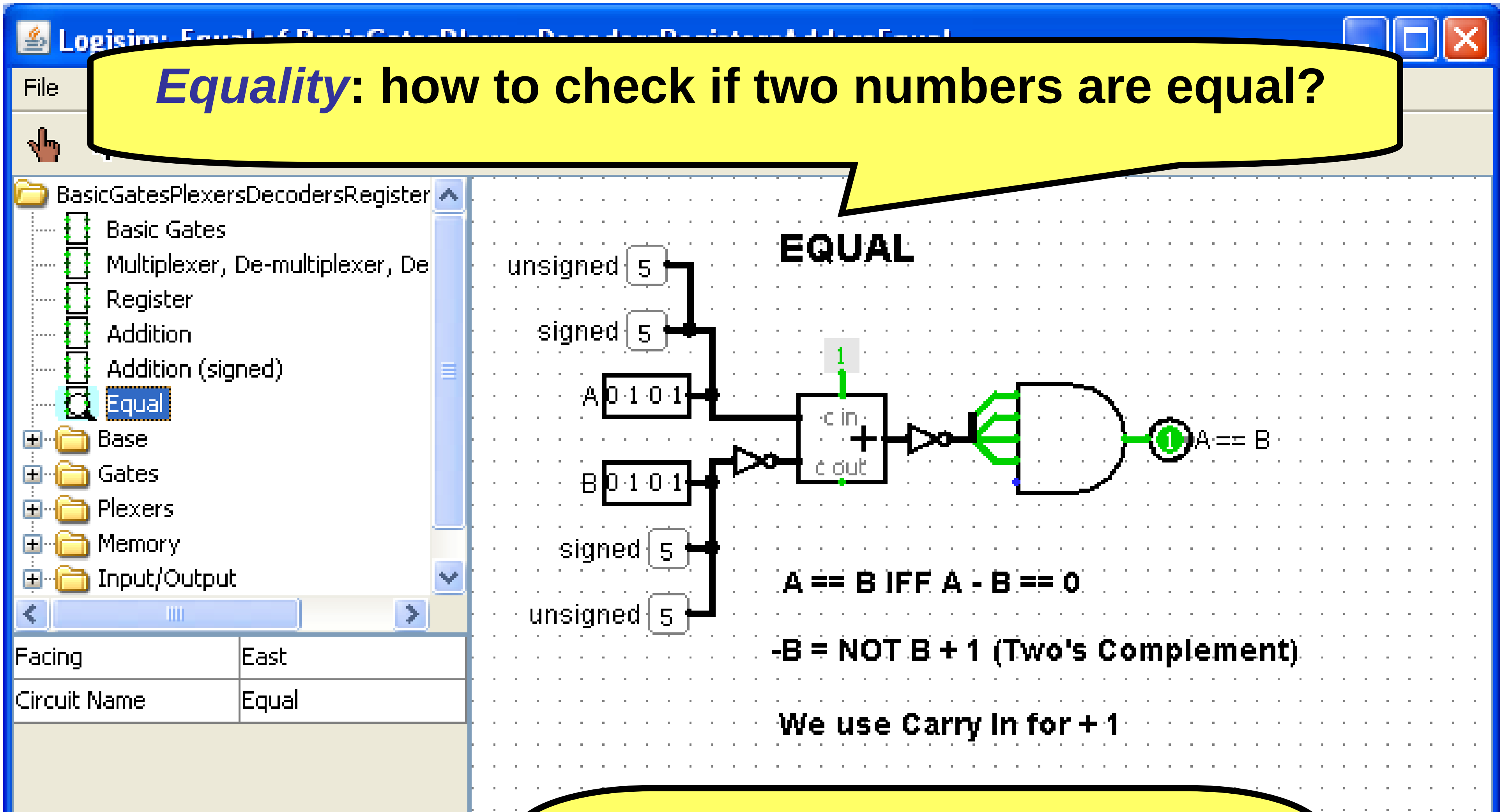
Signed integers - *Two’s Complement*



Facing	East
Output?	No
Bit Width	4
Th	No
	Unchanged
	B
	West
	SansSerif Plain 12

Remember: Using two’s complement subtraction becomes addition.

Equality: how to check if two numbers are equal?



EQUAL

A == B IFF A - B == 0

-B = NOT B + 1 (Two's Complement)

We use Carry In for + 1

Facing	East
Circuit Name	Equal

One way to check if two numbers are equal is to use the addtion circuit to subtract the numbers and check if the result is zero.

After this short introduction to digital logic you should have a basic understanding of:

Gates, Plexers, Decoders, Registers and Adders.

You also learned how to use *Logisim*.

These skills will be needed when you continue to construct more complex digital circuits.